第三章 DES加密算法

3.1 DES算法简介

- 目的

通信与计算机相结合是人类步入信息社会的一个阶梯，它始于六十年代末，完成于90年代。计算机通信网的形成与发展，要求信息作业标准化，安全保密亦不例外。只有标准化，才能真正实现网的安全，才能推广使用加密手段，以便于训练、生产和降低成本。

- 美国国家标准局(National Bureau of Standards, NBS)在1973年5月15公布了征求建议。1974年8月27日NBS再次出公告征求建议，对建议方案提出如下要求：
- 算法必须完全确定而无含糊之处；
- 算法必须有足够高的保护水准，即可以检测到威胁，恢复密钥所必须的运算时间或运算次数足够大；
- 保护方法必须只依赖于密钥的保密；
- 对任何用户或产品供应者必须是不加区分的。

- IBM公司在1971年完成的LUCIFER密码(64 bit分组，代换-置换，128 bit密钥)的基础上，改进成为建议的DES体制
- 1975年3月17日NBS公布了这个算法，并说明要以它作为联邦信息处理标准，征求各方意见。
- 1977年1月15日建议被批准为联邦标准[FIPS PUB 46]，并设计推出DES芯片。
- 1981年美国国家标准学会(American National Standards Institute, ANSI)将其作为标准，称之为DEA[ANSI X3.92]
- 1983年国际标准化组织(International Organization for Standardization, ISO)采用它作为标准，称作DEA-1

- NSA宣布每隔5年重新审议DES是否继续作为联邦标准，1988年（FIPS46-1）、1993年（FIPS46-2），1998年不再重新审批DES为联邦标准。
- 虽然DES已有替代的数据加密标准算法,但它仍是迄今为止得到最广泛应用的一种算法,也是一种最有代表性的分组加密体制。
- 1993年4月,Clinton政府公布了一项建议的加密技术标准,称作密钥托管加密技术标准EES(Escrowed Encryption Standard)。算法属美国政府SECRET密级。

- DES发展史确定了发展公用标准算法模式,而EES的制定路线与DES的背道而驰。人们怀疑有陷门和政府部门肆意侵犯公民权利。此举遭到广泛反对。
- 1995年5月AT&T Bell Lab的M. Blaze博士在PC机上用45分钟时间使SKIPJACK的LEAF协议失败,伪造ID码获得成功。1995年7月美国政府宣布放弃用EES来加密数据,只将它用于语音通信。
- 1997年1月美国国家标准技术研究所(National Institute of Standards and Technology, NIST)着手进行AES(Advanced Encryption Standard)的研究,成立了标准工作室。2001年Rijndael方案被批准为AES标准。

- DES(Data Encryption Standard)算法于1977年得到美国政府的正式许可，是一种用56位密钥来加密64位数据的方法。这是IBM的研究成果。
- DES是第一代公开的、完全说明细节的商业级现代密码算法，并被世界公认。

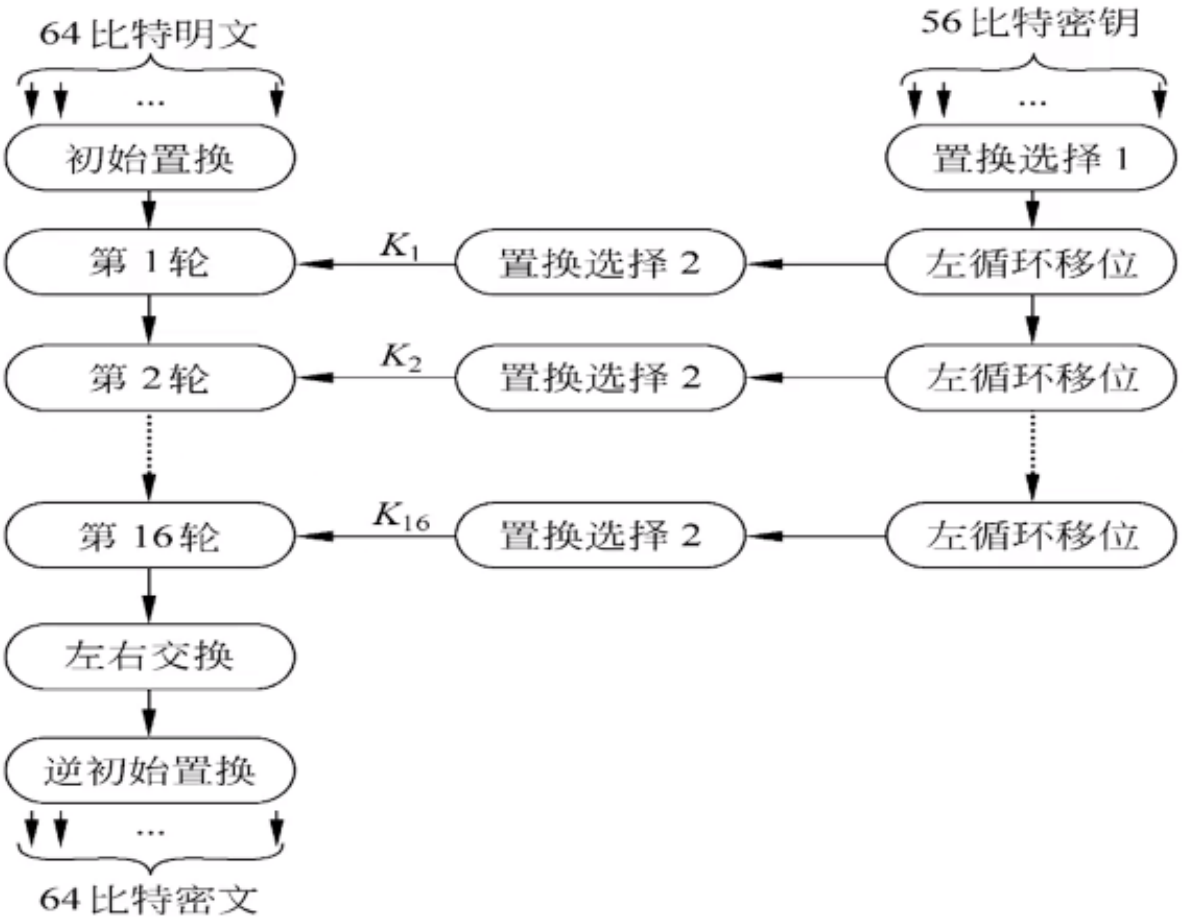
3.2 DES算法

3.2.1 DES的框架

- 分组长度为64 bits(8 bytes)。
- 密文分组长度也是64 bits。
- 密钥长度为64 bits，有8 bits奇偶校验，有效密钥长度为56 bits。
- 算法主要包括：初始置换IP、16轮迭代的乘积变换、逆初始置换IP^{-1}以及16个子密钥产生器。

3.2.1 DES的框架

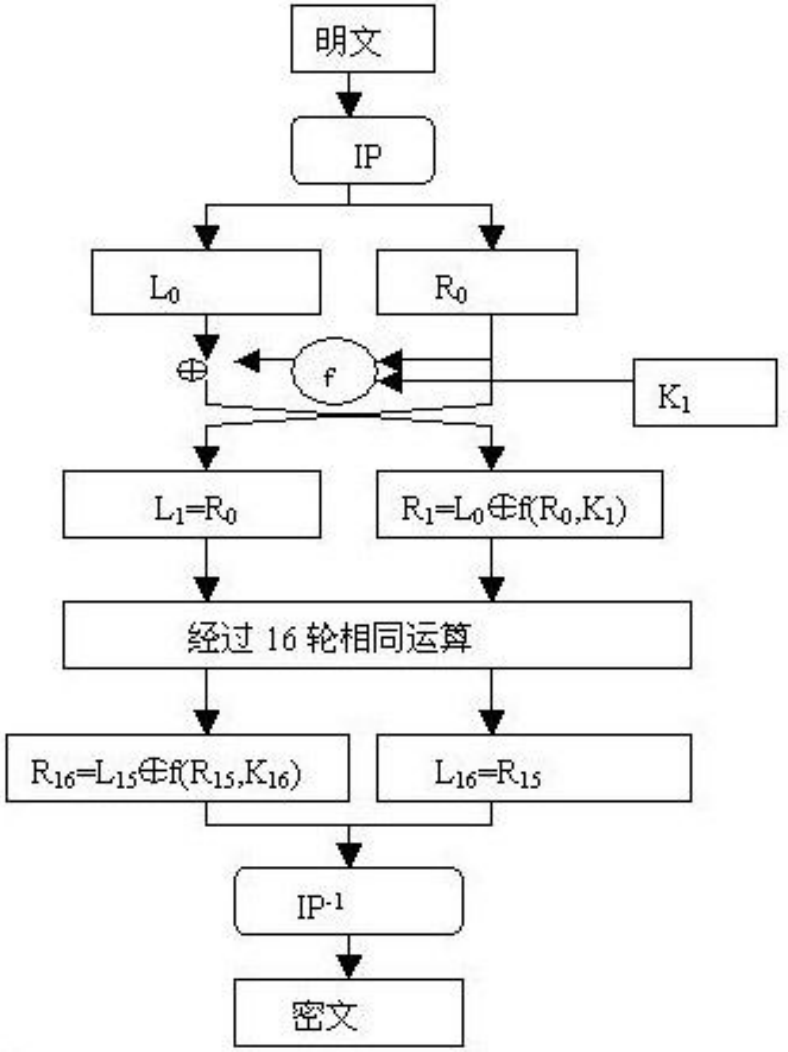
- DES算法框架如图所示：



3.2.1 DES的框架

- DES算法流程：

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$



3.2.2 初始置换IP与逆初始置换

- 初始置换是将64 bit明文的位置进行置换，得到一个乱序的64 bit明文组。
- 逆初始置换IP^{-1}。将16轮迭代后给出的64 bit组进行置换，得到输出的密文组。输出为阵中元素按行读得的结果。
- IP和IP^{-1}在密码意义上作用不大，它们的作用在于打乱原来输出x的ASCII码字划分的关系。

3.2.2 初始置换IP与逆初始置换

(a) 初始置换 IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

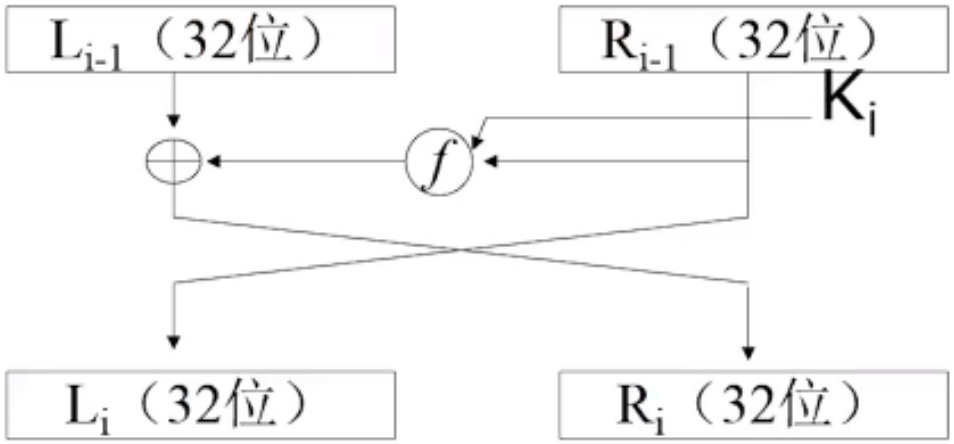
3.2.2 初始置换IP与逆初始置换

（b） 逆初始置换 IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

3.2.3 DES的轮函数

- DES轮函数的结构：



设输入为（x, y），　则DES的轮函数输出为：　$(y, x \oplus f_k(y))$

它等价于两个对合变换的复合：

$$(x, y) \mapsto (x \oplus f(k, y), y) \mapsto (y, x \oplus f(k, y))$$

3.2.3 DES的轮函数

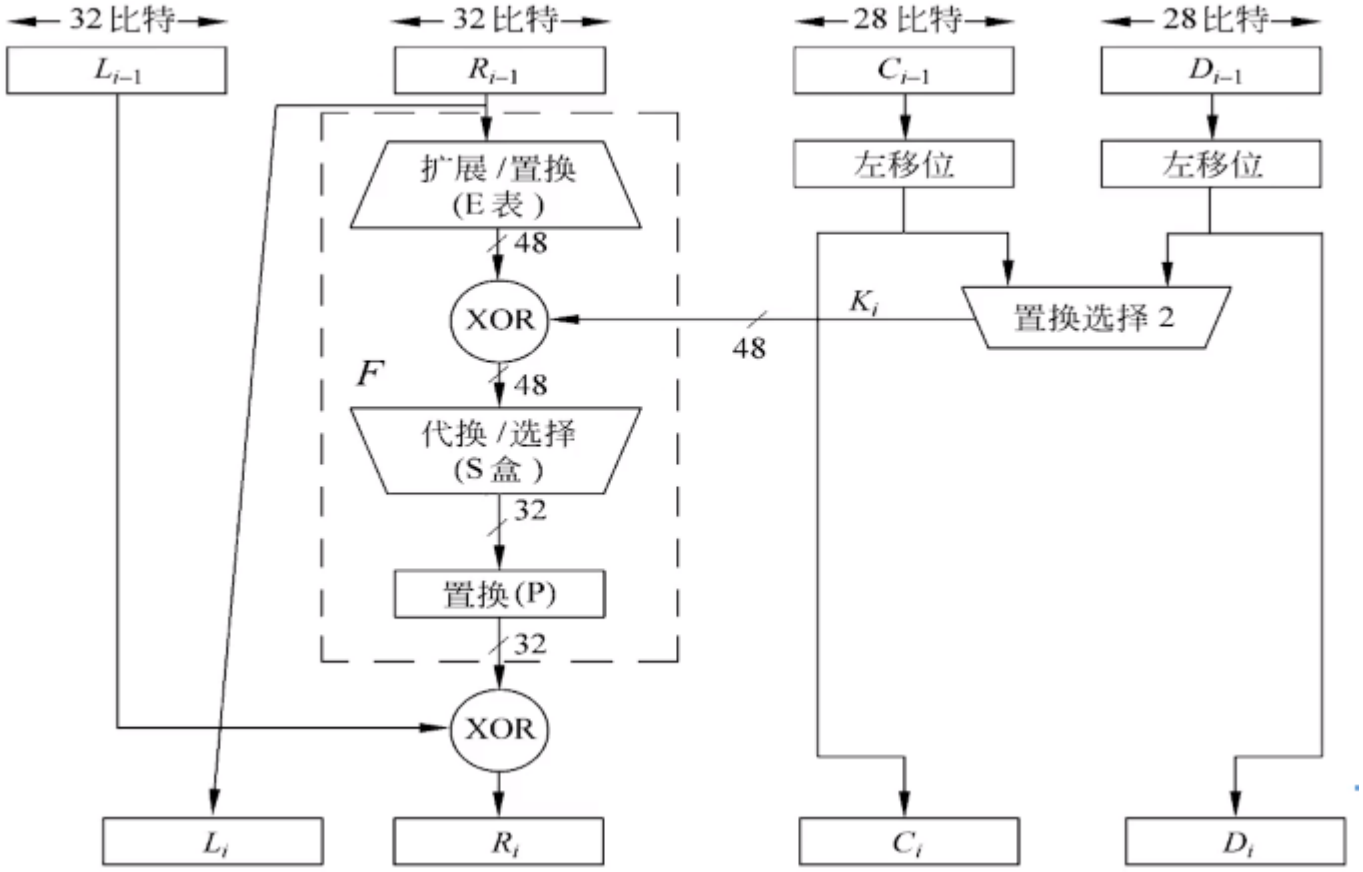
- 无论f函数如何选取，DES的轮函数是一个对合变换。

$$F(x,y)=(x\oplus f(k,y),y)$$

$$F(F(x,y))=F(x\oplus f(k,y),y)=((x\oplus f(k,y))\oplus f(k,y),y)=(x,y)$$

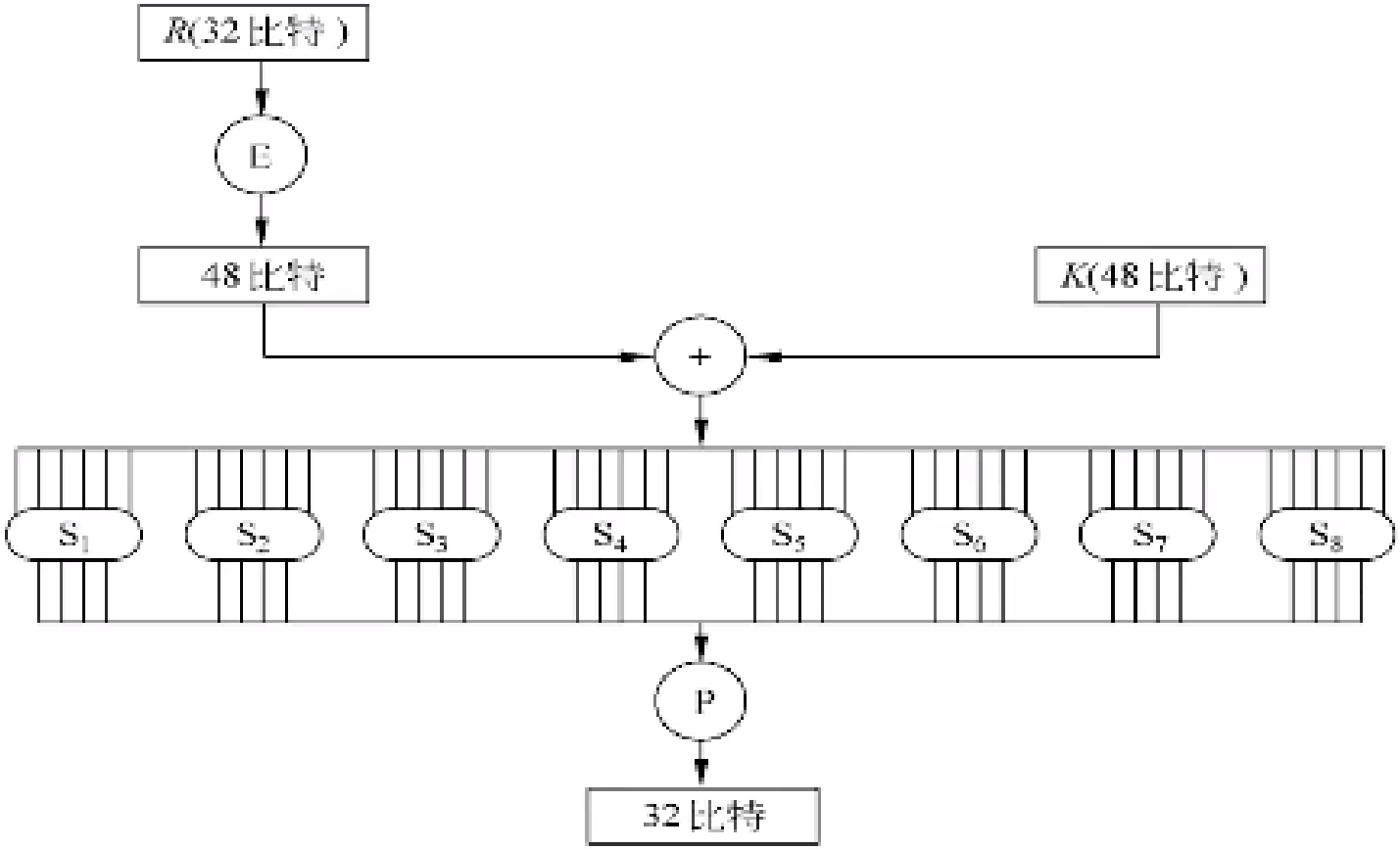
3.2.3 DES的轮函数

- DES算法轮结构如图所示：



3.2.3 DES的轮函数

- 函数f(R,K)的计算过程：



3.2.3 DES的轮函数

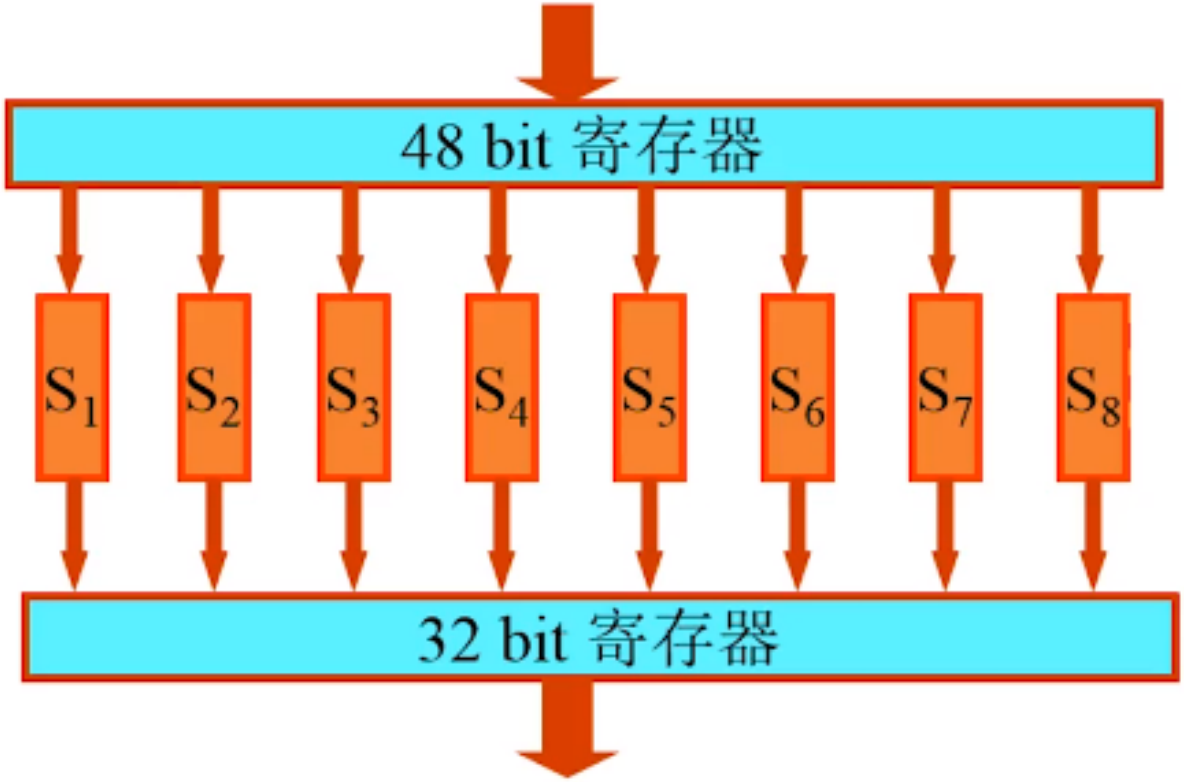
- 选择扩展运算E：

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**

注：扩展指的是位置的扩展而不是包含的比特值的扩展

3.2.3 DES的轮函数

- 选择压缩运算S：



3.2.3 DES的轮函数

- DES的S盒：

	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	S_1
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	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	

3.2.3 DES的轮函数

- DES的S盒的输入和输出关系：

$x_5\ x_0$ = 1 0　　$x_5\ x_4\ x_3\ x_2\ x_1\ x_0$ = 1 0 1 1 0 0

$(y_3, y_2, y_1, y_0)=(0,0,1,0)$

行号＼列号	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	2	14	10	0	6	13

3.2.3 DES的轮函数

将S-盒变换后的32比特数据再进行P盒置换，置换后得到的32比特即为f函数的输出。

- P盒置换的基本特点：

（1）P盒的各输出块的4个比特都来自不同的输入块；

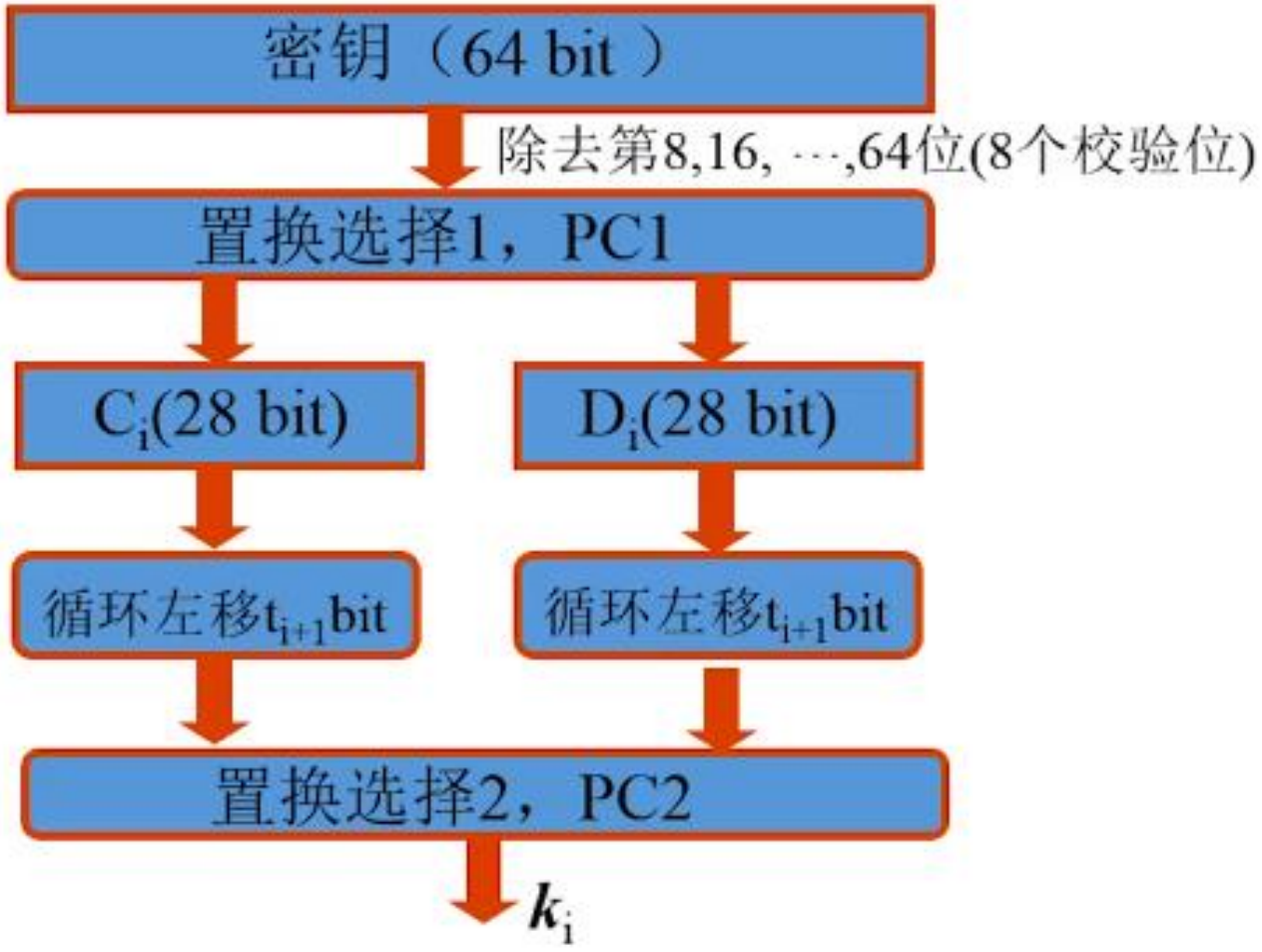
（2）P盒的各输入块的4个比特都分配到不同的输出块之中；

（3）P盒的第t输出快的4个比特都不来自第t输入块。

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

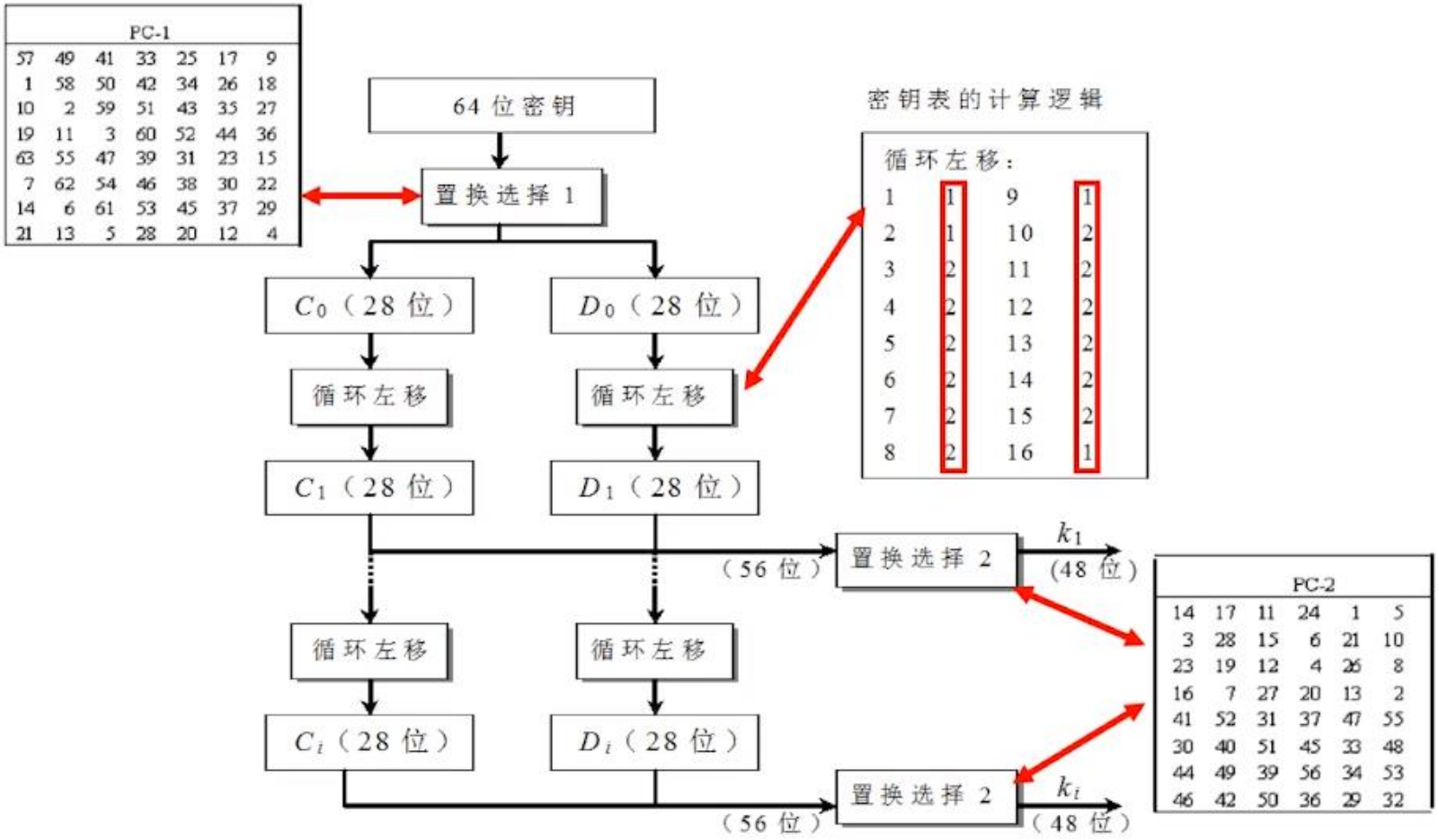
含义：P盒输出的第1个元是输入的第16个元

3.2.4 DES的密钥编排



3.2.4 DES的密钥编排

- DES中的子密钥的生成：



3.2.4 DES的密钥编排

- DES中的子密钥的生成：

PC-1						
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

PC-2					
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

移位次数表

第i次迭代	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
循环左移次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3.3 DES算法安全性

3.3.1 DES的弱密钥

- DES密钥的互补性。DES算法具有下属性质。若明文组x逐位取补，密钥k逐位取补，即$y=DES_k(x)$，则有 $\bar{y} = DES_k(\bar{x})$

  这种互补性会使DES在选择明文破译下所需的工作量减半。

- 弱密钥和半弱密钥

  （1）弱密钥：在同一密钥下加密两次为恒等变换，DES存在4个弱密钥。

  （2）半弱密钥：密钥对K1和K2互不相同，用K1加密的密文可以用K2来解密，至少有12个半弱密钥。

3.3.1 DES的弱密钥

- DES算法在每次迭代时都有一个子密钥供加密用。如果给定初始密钥k, 各轮的子密钥都相同，即有$k_1=k_2=...=k_{16}$, 就称给定的密钥k为弱密钥。

原始密钥	置换选择1后的密钥
	C D
(0, 0)⇌01 01 01 01 01 01 01 01	(0, 0)⇌00 00 00 00 00 00 00
(0,15)⇌1*F* 1*F* 1*F* 1*F* 0*E* 0*E* 0*E* 0*E*	(0,15)⇌00 00 00 0*F* *FF* *FF* *FF*
(0,15)⇌*E*0 *E*0 *E*0 *E*0 *F*1 *F*1 *F*1 *F*1	(0,15)⇌*FF* *FF* *FF* *F*0 00 00 00
(0,15)⇌*FE* *FE* *FE* *FE* *FE* *FE* *FE* *FE*	(0,15)⇌*FF* *FF* *FF* *FF* *FF* *FF* *FF*

3.3.2 DES密钥长度的争论

- DES算法公开发表以后，引起了一场激烈的争论
- 对DES安全性批评意见中，较为一致的看法是DES的密钥短了些。IBM最初向NBS提交的建议方案采用112位密钥，但公布的DES标准采用56位密钥，有人认为NSA故意限制DES的密钥长度。
- 采用穷搜索已经对DES构成了威胁。
- 1977年Diffie和Hellman提出了制造一个每秒能测试106个密钥的大规模芯片，这种芯片的机器大约一天就可以搜索DES算法的整个密钥空间，制造这样的机器需要两千万美元。

3.3.2 DES密钥长度的争论

- 1993年，R.Session和M.Wiener给出了一个非常详细的密钥搜索机器的设计方案
- 基于并行的密钥搜索芯片，此芯片美妙测试5×107个密钥
- 当时这种芯片的造价是10.5美元，5760个这样的芯片组成的系统需要10万美元，这一系统平均1.5天就可以找到密钥
- 如果利用10个这样的系统，费用是100万美元，但搜索时间可以降到2.5个小时。

3.3.2 DES密钥长度的争论

- DES的56位短密钥面临的另外一个严峻而现实的问题是：国际互联网Internet的超级计算能力。
- 1997年1月28日，美国的RSA数据安全公司在互联网上开展了一项名为“密钥挑战”的竞赛，悬赏一万美元，破解一段用56位密钥加密的DES密文。

3.3.2 DES密钥长度的争论

- 一位名叫Rocke VerSer的程序员设计了一个可以通过互联网分段运行的密钥穷举搜索程序，组织实施了一个称为DESIIALL的搜索行动，成千上万的志愿者加入到计划中。
- 计划实施的第96天，即挑战赛计划公布的第140天，1997年6月17日晚上10点39分，美国盐湖城Inetz公司的职员Michael Sanders成功地找到了密钥。
- 在计算机上显示了明文：“The unknown message is: Strong cryptography makes the world a safer place”

3.3.2 DES密钥长度的争论

- 1998年7月电子前沿基金会（Electronic Frontier Foundation, EFF）使用一台25万美元的电脑在56小时内破译了56比特密钥的DES。
- 1999年1月RSA数据安全会议期间，电子前沿基金会用22小时15分钟就宣告破解了一个DES的密钥

3.3.2 DES密钥长度的争论

- DES的其他攻击方法

目前攻击DES的主要方法有时间-空间权衡攻击、差分攻击、线性攻击和相关密钥攻击等方法，在这些攻击方法中，线性攻击方法是最有效的一种方法。

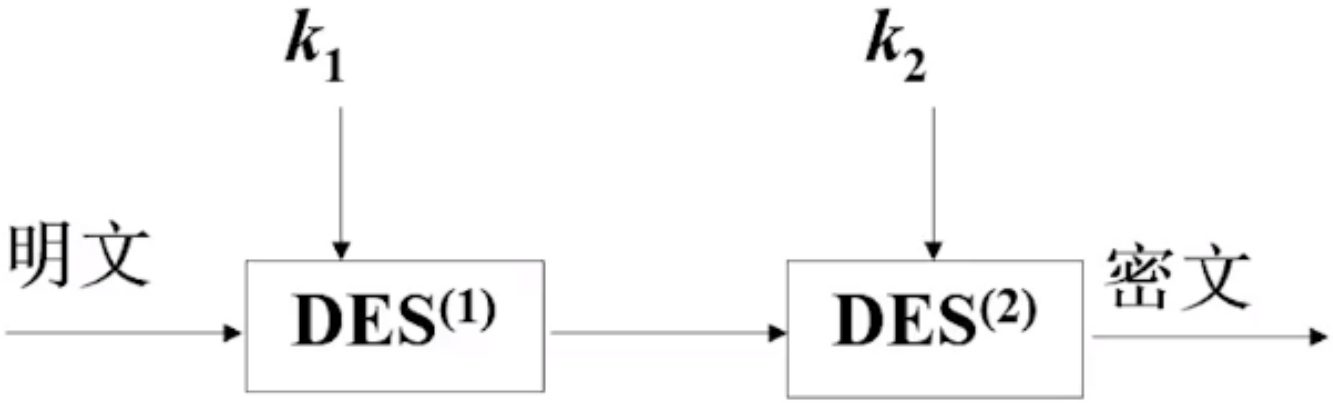
3.4 3DES加密算法

3.4.1 多重DES

- 如果一个分组密码易受到穷举密钥搜索攻击，那么对同一消息多次加密就有可能增强安全性。
- 多重DES就是使用多个密钥利用DES对明文进行多次加密。使用多重DES可以增加密钥量，从而大大提高抵抗穷举密钥搜索攻击的能力。
- 多重加密类似于一个有着多个相同密码的级联，但各级密码无需独立，且每级密码既可以是一个分组密码加密函数，也可以相应的解密函数。

3.4.2 双重DES

- 简单的对消息x_i利用两个不同的密钥进行两次加密
- 目的是为了抵抗穷搜索攻击，期望密钥长度扩展为112比特



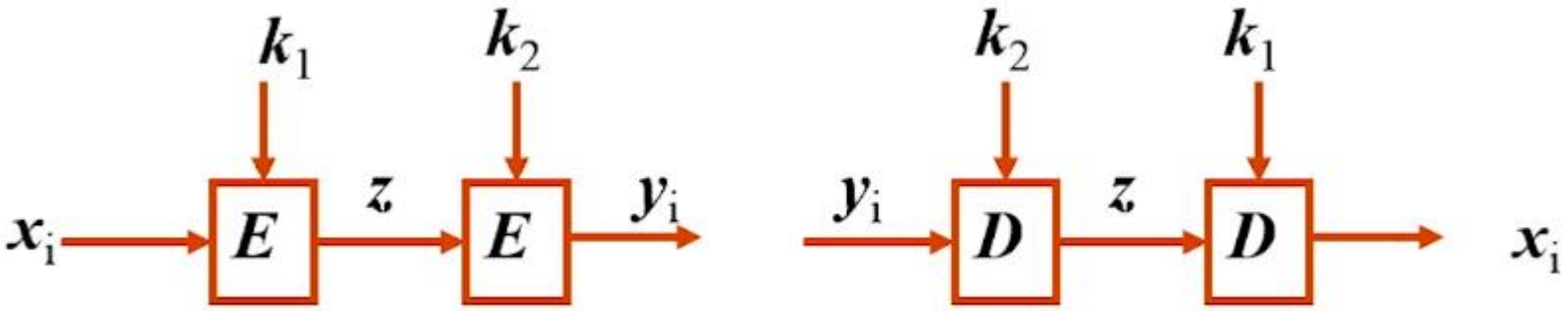
3.4.2 双重DES

- 中间相遇攻击

由Diffie和Hellman最早提出，可以降低搜索量，基本想法如下：

若有明文/密文对(x_i, y_i)满足：$y_i = E_{k2}[E_{k1}(x_i)]$

则可得：$z = E_{k1}(x_i) = D_{k2}(y_i)$



3.4.2 双重DES

- 中间相遇攻击的步骤

给定一已知明密文(x_1, y_1)，可按下述方法攻击：

（1）以密钥k_1的所有2^{56}个可能的取值对此明文x_1加密，并将密文z存储在一个表中；

（2）从所有可能的2^{56}个密钥k_2中以任意次序选出一个对给定的密文y_1进行解密，并将每次解密结果z在上述表中查找相匹配的值。一旦找到，则可确定出两个密钥k_1和k_2；

（3）以此对密钥k_1和k_2对另一已知明文密文对(x_2, y_2)中的x_2进行加密，如果能得出相应的密文y_2就可以确定k_1和k_2是所要找的密钥。

3.4.2 双重DES

- 中间相遇攻击的复杂度

（1）对于给定明文x，以两重DES加密将有2^{64}个可能的密文；

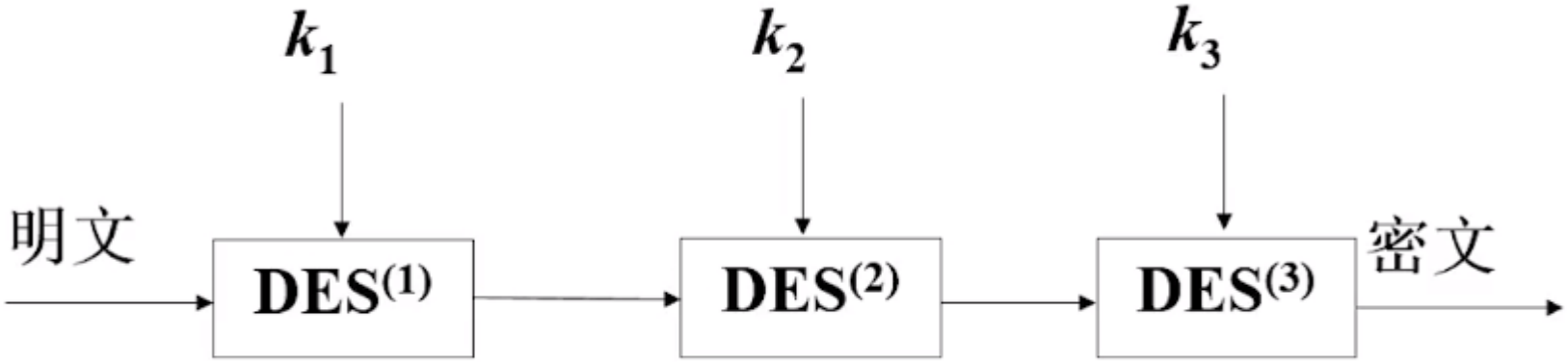
（2）可能的密钥数为2^{112}个。所以，在给定明文下，将有$2^{112}/2^{64} = 2^{48}$个密钥能产生给定的密文；

（3）用另一对64比特明文/密文对进行校验，就使虚报率降为$2^{48}-2^{64} = 2^{-16}$。

（4）这一攻击法所需的存储量为$2^{56}\times 8$ Byte，最大试验的加密次数$2\times 2^{56} = 2^{57}$。这说明破译双重DES的难度为liang2^{57}量级。

3.4.3 三重DES

- 三重DES中三个密码组件既可以是一个加密函数，也可以是一个解密函数。
- 当$k_1=k_3$时，则称为双密钥三重DES



3.4.3 三重DES

- 双密钥三重DES算法

  加密：$y = E_{k1}[D_{k2}[E_{k1}(x)]]$

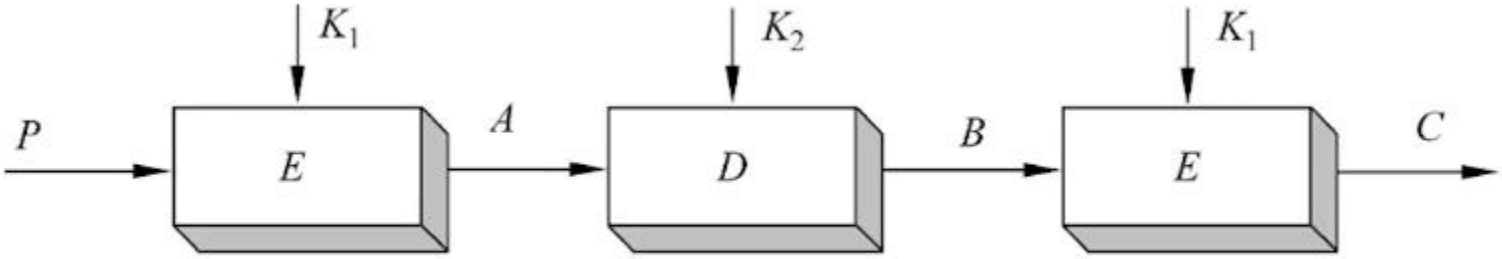
  解密：$x = D_{k1}[E_{k2}[D_{k1}(y)]]$

  称其为加密-解密-加密方案，简记为EDE。

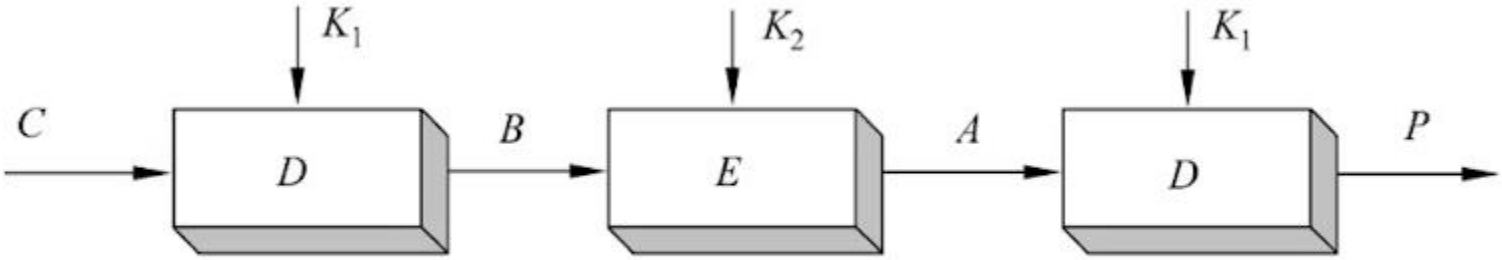
  此方案已在ANSI X9.17和ISO 8732标准中采用，并在保密增强邮件（PEM）系统中得到利用。

3.4.3 三重DES

- 双密钥三重DES算法



(a) 加密



(b) 解密

3.4.3 三重DES

- 双密钥三重DES算法的安全性
  - (1) 破译它的穷举密钥搜索量为$2^{112} \approx 5 \times 10^{35}$量级
  - (2) 差分攻击破译也要超过10^{52}量级
  - (3) 此方案仍有足够的安全性

问题：实现DES算法。