第三章作业

1．(1)设 M'是 M 的逐比特取反，证明：若 $Y=\mathrm{DES}_K(X)$则 $Y'=\mathrm{DES}_{K'}(X')$

证：①以 P_D 记 DES 中的所有置换，包括循环移位、左右交换，则 P_D 满足如下性质：

若 $T=P_D(Z)$，则 $T'=P_D(Z')$

在 DES 中，异或运算显然满足性质 $a'\oplus b'=a\oplus b$，及 $a'\oplus b=(a\oplus b)'$

因而 DES 中的函数 $F(R_{i-1}, K_i)$在 S 盒前是异或运算，所以 $F(R'_{i-1}, K'_i)=F(R_{i-1}, K_i)$

②由密钥编排方案中的运算部件知，

若 K 的子密钥为 $K_1,K_2,\ldots,K_{16}$，那么 K′的子密钥为 $K'_1,K'_2,\ldots,K'_{16}$

③若 X 经初始置换 IP 后记为 $L_0\|R_0$，则 X′经初始置换 IP 后记为 $L'_0\|R'_0$，

用 K 对 X 加密的第 i 轮输入为 $L_{i-1}\|R_{i-1}$，输出为 $L_i\|R_i$，

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

设用 K′对 X′加密的第 i 轮输入为 $L'_{i-1}\|R'_{i-1}$，则其第 i 轮的输出满足

左半部分$=R'_{i-1}=L'_i$

右半部分$=L'_{i-1}\oplus F(R'_{i-1}, K'_i)=L'_{i-1}\oplus F(R_{i-1}, K_i)=(L_{i-1}\oplus F(R_{i-1}, K_i))'=R'_i$

即 $L'_i\|R'_i$，

④由归纳法知，前 16 轮的输出均满足逐比特取反的关系，在经过左右交换盒 IP^{-1} 两个置换运算，输出密文也满足取反关系　　#

(2)由(1)的结论，在对 DES 进行穷搜索攻击时，选择两个明密文对(M,C_1)和(M',C_2)，然后选择 $K\in F_2^{56}$，对 M 加密 $C=\mathrm{DES}_K(M)$，判断 $C=C_1$ 或 $C'=C_2$ 则分别说明 K 或 K′为正确密钥，否则 K 和 K′都不是密钥，从而一次加密运算可同时验证一对互反密钥，使搜索量减少一半

2．证明：DES 的解密变换是加密变换的逆

证：DES 的加密变换由 IP，16 轮迭代，左右交换，IP^{-1} 四部分构成，注意到解密时子密钥逆续使用，16 轮迭代与左右交换一起刚好构成 Feistel 网络，若 Feistel 网络输入为 X，输出为 Y，即 Y＝Feistel(X,K)，其中 K 为密钥，如果 K 的子密钥逆续使用则记为 Inv(K)，那么由 Feistel 网络的性质有 X=Feistel(Y,Inv(K))。

对任意的明文消息 M 加密可表示为 $C=IP^{-1}[\mathrm{Feistel}(IP(M),K)]$

即 IP(C)=Feistel(IP(M),K)

由 Feistel 网络的性质有 IP(M)=Feistel(IP(C),Inv(K))

而对密文 C 解密，即为明文$=IP^{-1}[\mathrm{Feistel}(IP(C),\mathrm{Inv}(K))]=IP^{-1}[IP(M)]=M$，所以解密变换是加密变换的逆　　#

3．在 DES 的 CBC 模式中 C_1 的一个错误明显地将影响 P_1 和 P_2 的结果

（1）P_2 以后的分组不受影响，这是因为 C_1 以后的密文都是正确的，而恢复明文主要看对应密文分组和其前一个密文分组的正确性。

（2）加密前的明文分组 P_1 有 1 比特错误，则这一错误将在所有后续密文分组中传播，但接受者能够正确解密，除了 P_1 的一个错误比特之外。这是因为相当于发送者将明文改变了 1 比特得到一个新明文，而该明文的对应密文正确的传送给了接受方。

4．在 8bitCFB 中密文字符中出现 1 比特错误，该错误将影响包括该密文的连续 9 组密文的解密。

5．在实现 IDEA 时，最困难的部分是模 $2^{16}+1$ 乘法运算，设 a 和 b 是两个 n 比特的非 0 整数，记$(ab \bmod 2^n)$为 ab 的 n 个最低有效位，$(ab \operatorname{div} 2^n)$为 ab 右移 n 位

(1) 证明存在惟一的非负整数 q 和 r，使得 $ab=q(2^n+1)+r$

证：令 q 为 $ab/(2^n+1)$的商，r 为 $ab/(2^n+1)$的余数，均非负，则 $ab=q(2^n+1)+r$

若存在另一对数 q_1，r_1 满足 $ab=q_1(2^n+1)+r_1$

两式相减的$(r_1-r)=(q-q_1)(2^n+1)$

由于$|(r_1-r)|\le 2^n$，所以当且仅当 $r_1=r$，$q_1=q$ 时上式才成立　#

(2) 求 q 和 r 的上下界

解：由(1)知，r 为余数，所以有 $0\le r\le 2^n$

又 a 和 b 的最大值为 2^n-1，所以 $0\le q\le [ab/(2^n+1)]\le [(2^n-1)^2/(2^n-1)]=[2^n-3+4/(2^n+1)]=2^n-3$ 对于 $n\ge 2$ 时都成立，当 n＝1 时，q＝0

所以，q 的上下界为 $0\le q\le 2^n-3$ （$n\ge 2$）n＝1 时，q＝0

(3) 证明 $q+r<2^{n+1}$

证：$q+r\le 2^n-3+2^n=2^{n+1}-3<2^{n+1}$

(4)，(5) 求(ab div 2^n)关于 q 的表达式和(ab mod 2^n)关于 q 和 r 的表达式

解：设 $ab=q_1 2^n+r_1$，则显然有 $q_1=(ab \text{ div } 2^n)$，$r_1=(ab \bmod 2^n)$

又 $ab=q(2^n+1)+r$，记为(a)式

当 $q_1\le r_1$ 时，$ab=q_1 2^n+r_1=q_1(2^n+1)+(r_1-q_1)$　记为(b)式

此时 $0\le (r_1-q_1)\le r_1\le 2^n-1$

由(1)的唯一性结论,比较(a),(b)两式知，$q_1=q$，$r_1-q_1=r$

即 $q_1=(ab \text{ div } 2^n)=q$，$r_1=(ab \bmod 2^n)=r+q_1=r+q$

当 $q_1>r_1$ 时，$ab=q_1 2^n+r_1=(q_1-1)(2^n+1)+(r_1-q_1)+(2^n+1)$　记为(c)式

此时，由假设 $q_1>r_1$，知$(r_1-q_1)<0$，及$(q_1-1)\ge 0$ 所以

$0<(r_1-q_1)+(2^n+1)\le 2^n$

即$(r_1-q_1)+(2^n+1)$为 $ab/(2^n+1)$的余数

所以比较两式(a),(c)两式知$(q_1-1)=q$，$(r_1-q_1)+(2^n+1)=r$

即 $q_1=(ab \text{ div } 2^n)=q+1$，$r_1=(ab \bmod 2^n)=r+q_1-(2^n+1)=r+(q+1)-(2^n+1)$

综上所述有

$$q_1=(ab \text{ div } 2^n)=\begin{cases} q, & q_1\le r_1 \\ q+1, & q_1\le r_1 \end{cases}=\begin{cases} q, & (ab\ div\ 2^n)\le (ab \bmod 2^n) \\ q+1, & (ab\ div\ 2^n)\ge (ab \bmod 2^n) \end{cases}$$

$$r_1=(ab \bmod 2^n)=\begin{cases} q+r, & q_1\le r_1 \\ q+r-2^n, & q_1\le r_1 \end{cases}=\begin{cases} q+r, & (ab\ div\ 2^n)\le (ab \bmod 2^n) \\ q+r-2^n, & (ab\ div\ 2^n)\ge (ab \bmod 2^n) \end{cases}$$

(6) 用(4)和(5)的结果求 r 的表达式，说明 r 的含义

当 $q_1\le r_1$ 时，$r=(r+q)-q=(r_1-q_1)=(ab \bmod 2^n)-(ab \text{ div } 2^n)$

当 $q_1>r_1$ 时，　$r=(r+q-2^n)-(q+1)+(2^n+1)=(r_1-q_1)+(2^n+1)=(ab \bmod 2^n)-(ab \text{ div } 2^n)+(2^n+1)$

所以 $r=ab \bmod (2^n+1)=\begin{cases} ab \bmod 2^n-ab\ div\ 2^n, & (ab\ div\ 2^n)\le (ab \bmod 2^n) \\ ab \bmod 2^n-ab\ div\ 2^n+2^n+1, & (ab\ div\ 2^n)\ge (ab \bmod 2^n) \end{cases}$

6. （1）在 IDEA 模乘运算中，将模数取为 $2^{16}+1$，是因为它是素数，从而所有非 0 元都有逆元

(2) 在 IDEA 的模加运算中，模数取为 2^{16} 使所有元素都有逆元，构成群运算，同时刚好在 16 位子段上运算，求模运算易于实现，而取为 $2^{16}+1$ 时则必须做额外处理