密码编码学与网络安全-原理与实践（第五版）课后习题答案

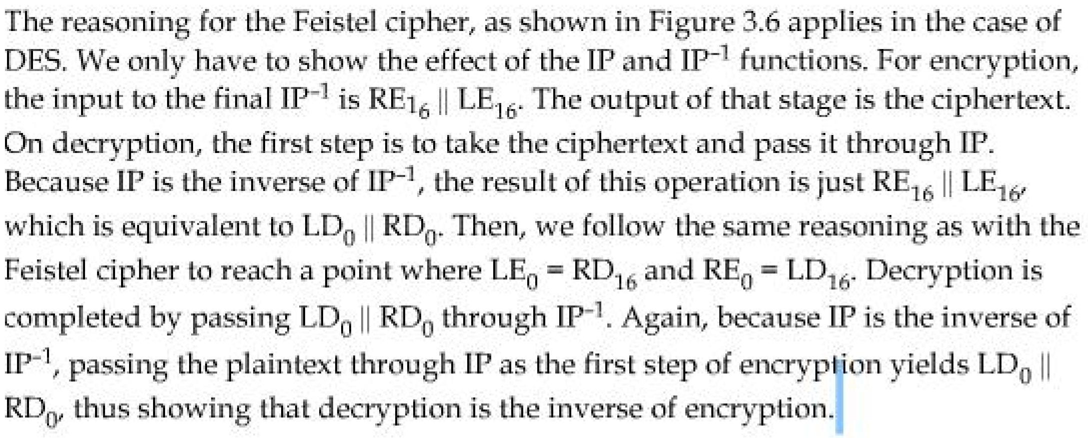
2.17

答案：PBVWETLXOZR

3.9　证明 DES 解密算法实际上是加密算法的逆

思路：　逆运算就是 A=*f*(B)，B=*g*(A)，*f*、*g* 互为逆运算，加解密就是这个特点否则解密出来的东西和明文不一样怎么能够加解密呢？　做的就是去证明为什么密文解密为什么会得到明文。

答案一：



The reasoning for the Feistel cipher, as shown in Figure 3.6 applies in the case of DES. We only have to show the effect of the IP and IP^{-1} functions. For encryption, the input to the final IP^{-1} is $RE_{16} \| LE_{16}$. The output of that stage is the ciphertext. On decryption, the first step is to take the ciphertext and pass it through IP. Because IP is the inverse of IP^{-1}, the result of this operation is just $RE_{16} \| LE_{16}$, which is equivalent to $LD_0 \| RD_0$. Then, we follow the same reasoning as with the Feistel cipher to reach a point where $LE_0 = RD_{16}$ and $RE_0 = LD_{16}$. Decryption is completed by passing $LD_0 \| RD_0$ through IP^{-1}. Again, because IP is the inverse of IP^{-1}, passing the plaintext through IP as the first step of encryption yields $LD_0 \|$ RD_0, thus showing that decryption is the inverse of encryption.

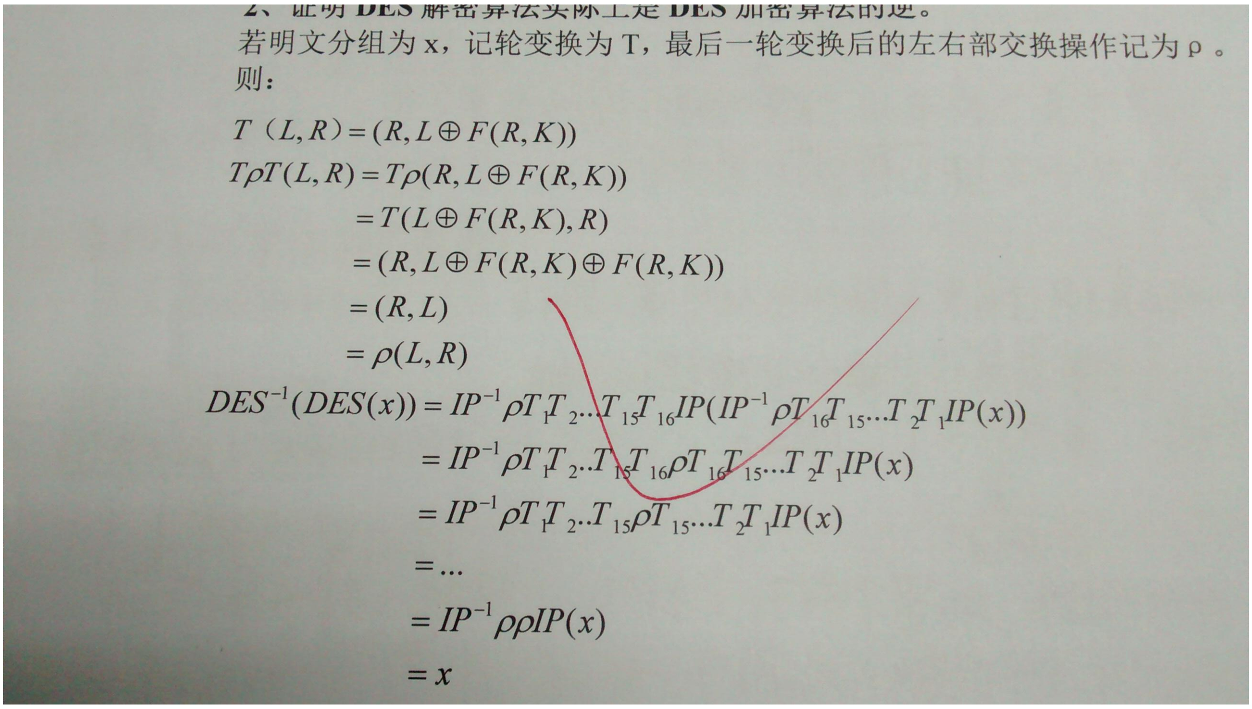
注：

1. 倒数第三行应为 $LE_0||RE_0$，倒数第二行和第一行应为 $LE_0||RE_0$

由于 IP^{-1}（$LE_{17}||RE_{17}$）$=IP^{-1}$（$RE_{16}||LE_{16}$）= C， 故 $RE_{16}||LE_{16} = LE_{17}||RE_{17} = IP$（C）

由于 $RD_{16}||LD_{16} = LE_0||RE_0 = IP$（M）， 故 IP^{-1}（$LE_0||RE_0$）= M

答案二：



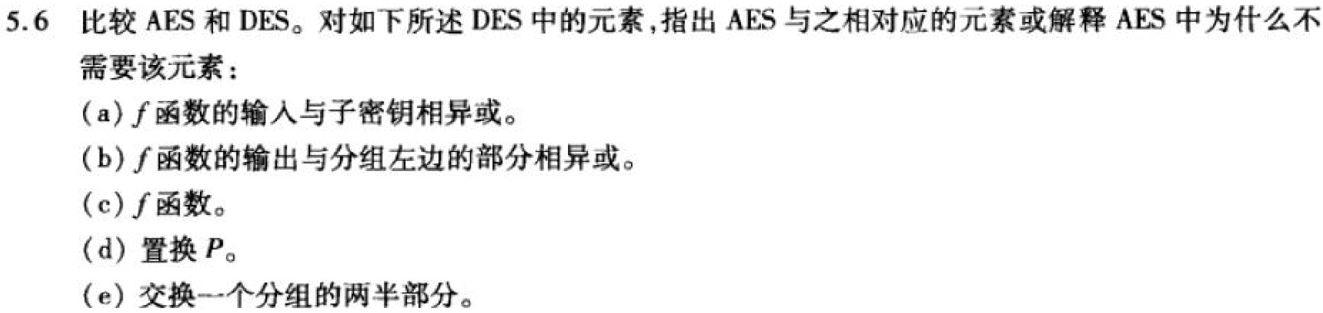
2、证明 **DES** 解密算法实际上是 **DES** 加密算法的逆。

若明文分组为 x，记轮变换为 T，最后一轮变换后的左右部交换操作记为 ρ 。则：

$$T(L,R)=(R,L\oplus F(R,K))$$

$$\begin{aligned}T\rho T(L,R)&=T\rho(R,L\oplus F(R,K))\\&=T(L\oplus F(R,K),R)\\&=(R,L\oplus F(R,K)\oplus F(R,K))\\&=(R,L)\\&=\rho(L,R)\end{aligned}$$

$$\begin{aligned}DES^{-1}(DES(x))&=IP^{-1}\rho T_1T_2..T_{15}T_{16}IP(IP^{-1}\rho T_{16}T_{15}...T_2T_1IP(x))\\&=IP^{-1}\rho T_1T_2..T_{15}T_{16}\rho T_{16}T_{15}...T_2T_1IP(x)\\&=IP^{-1}\rho T_1T_2..T_{15}\rho T_{15}...T_2T_1IP(x)\\&=...\\&=IP^{-1}\rho\rho IP(x)\\&=x\end{aligned}$$



5.6 比较 AES 和 DES。对如下所述 DES 中的元素，指出 AES 与之相对应的元素或解释 AES 中为什么不需要该元素：

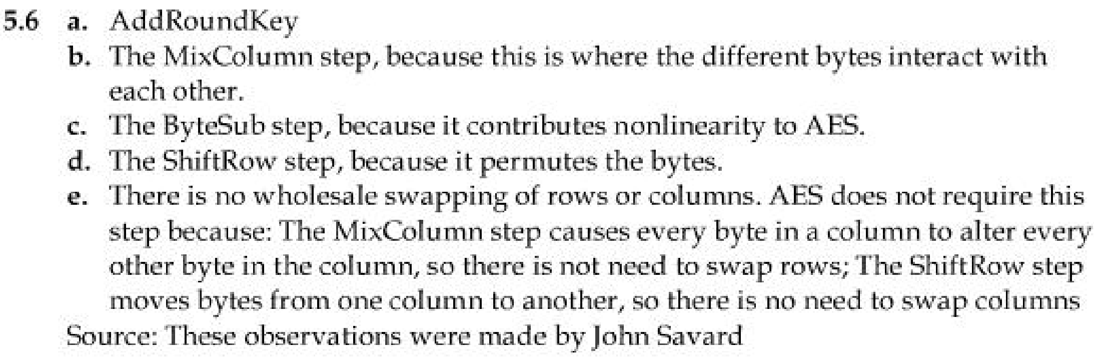
(a) *f* 函数的输入与子密钥相异或。

(b) *f* 函数的输出与分组左边的部分相异或。

(c) *f* 函数。

(d) 置换 *P*。

(e) 交换一个分组的两半部分。



5.6 **a.** AddRoundKey

b. The MixColumn step, because this is where the different bytes interact with each other.

c. The ByteSub step, because it contributes nonlinearity to AES.

d. The ShiftRow step, because it permutes the bytes.

e. There is no wholesale swapping of rows or columns. AES does not require this step because: The MixColumn step causes every byte in a column to alter every other byte in the column, so there is not need to swap rows; The ShiftRow step moves bytes from one column to another, so there is no need to swap columns

Source: These observations were made by John Savard

9.3

解： n=35 -> p=5, q=7

ϕ(n)=(p-1)(q-1)=24

$d\equiv e^{-1} \bmod \phi(n)\equiv 5^{-1} \bmod 24\equiv 5 \bmod 24$ .... (因为 $5\times5\equiv1 \bmod 24$)

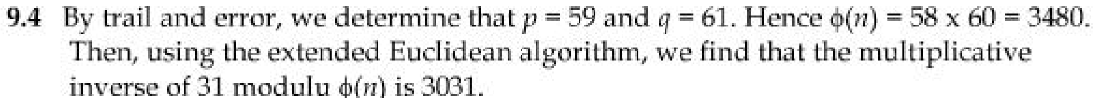
所以，明文 $M \equiv C^{d} \bmod n \equiv 10^{5} \bmod 35 \equiv 5$

快速指数算法求模幂 10^5 mod 35： a^m mon n m=$(bi)_2$

$5 = 4 + 1 = (101)_2$ b_i=0, d←d*d mod n

b_i - 1 0 1 b_i=1, d←d*d*a mod n

d 1 10 30 5



9.4 By trail and error, we determine that p = 59 and q = 61. Hence $\phi(n)$ = 58 x 60 = 3480. Then, using the extended Euclidean algorithm, we find that the multiplicative inverse of 31 modulu $\phi(n)$ is 3031.

9.4

法二： 由欧拉定理，得 e^(φ（φ(n)))=1 (mod φ(n)),

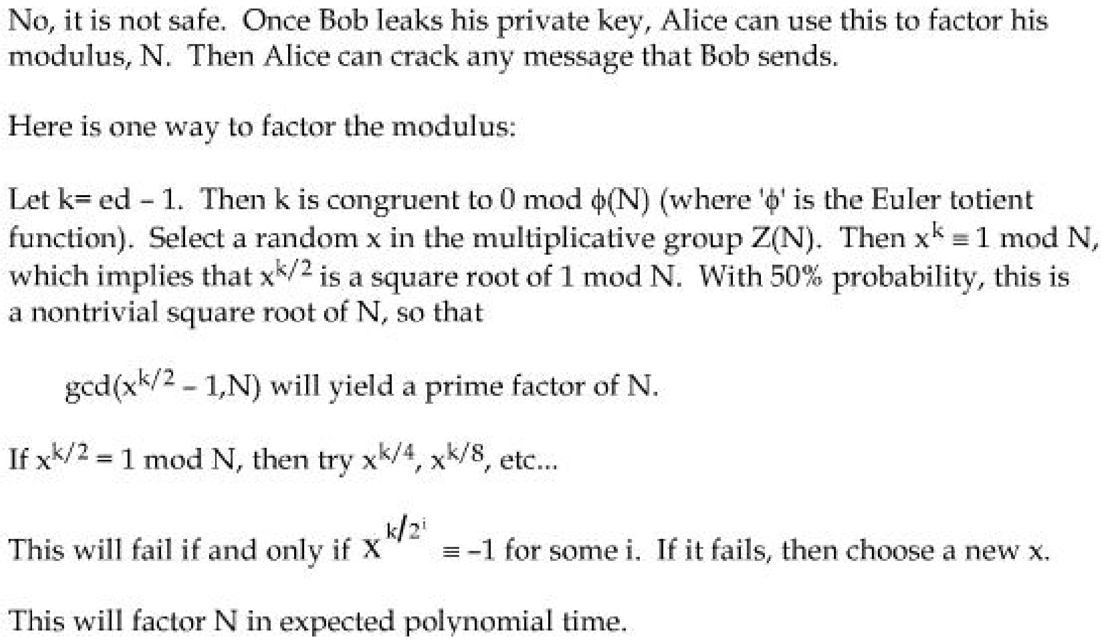
从而 d= e^(φ（φ(n))-1) (mod φ(n))

由于 φ（n)=58×60=3480 =2^3 × 3 × 5 × 29，

故 φ（3480）= （2^3-4）× 2 × 4 × 28=896

从而 d= 31^{895} （mod 3480) = 3031

9.7



No, it is not safe. Once Bob leaks his private key, Alice can use this to factor his modulus, N. Then Alice can crack any message that Bob sends.

Here is one way to factor the modulus:

Let k= ed – 1. Then k is congruent to 0 mod ϕ(N) (where 'ϕ' is the Euler totient function). Select a random x in the multiplicative group Z(N). Then $x^k \equiv 1$ mod N, which implies that $x^{k/2}$ is a square root of 1 mod N. With 50% probability, this is a nontrivial square root of N, so that

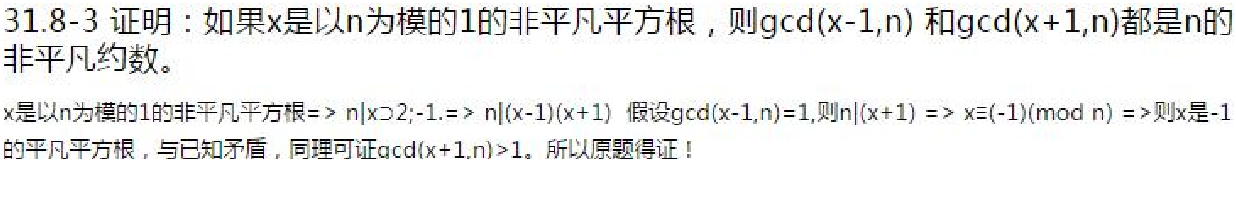
gcd($x^{k/2}$ – 1,N) will yield a prime factor of N.

If $x^{k/2}$ = 1 mod N, then try $x^{k/4}$, $x^{k/8}$, etc...

This will fail if and only if $X^{k/2^i} \equiv -1$ for some i. If it fails, then choose a new x.

This will factor N in expected polynomial time.

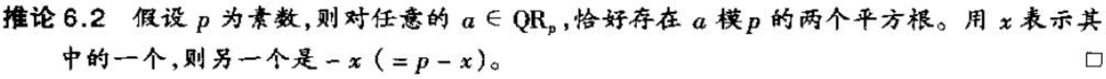
（注：根据以下事实：



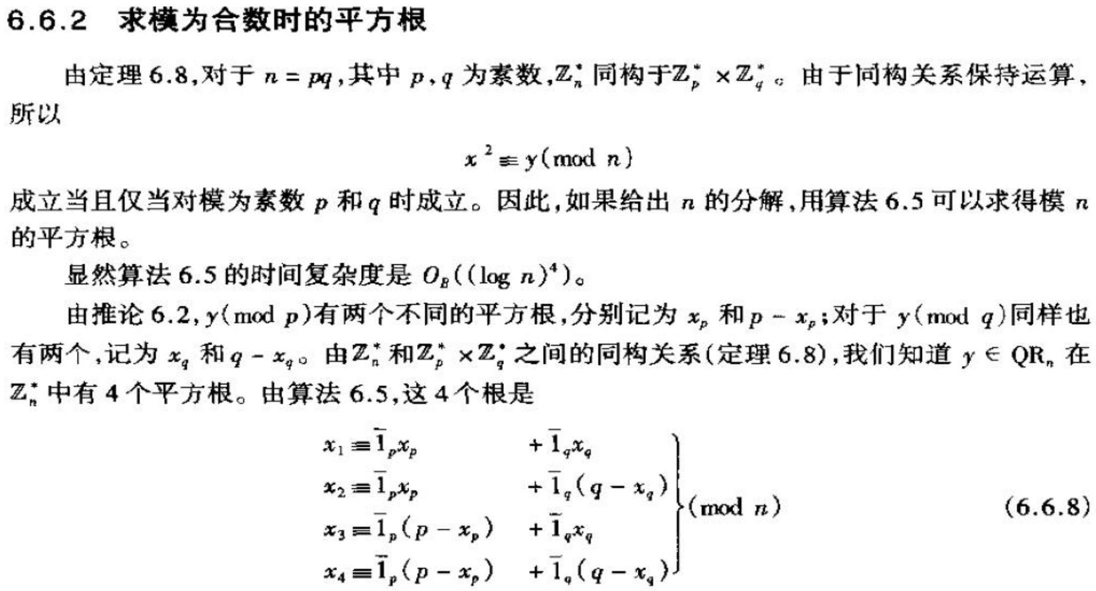
31.8-3 证明：如果x是以n为模的1的非平凡平方根，则gcd(x-1,n) 和gcd(x+1,n)都是n的非平凡约数。

x是以n为模的1的非平凡平方根=> n|x⊃2;-1.=> n|(x-1)(x+1) 假设gcd(x-1,n)=1,则n|(x+1) => x≡(-1)(mod n) =>则x是-1的平凡平方根，与已知矛盾，同理可证gcd(x+1,n)>1。所以原题得证！

毛文波《现代密码学理论与实践》（电子版可从网上找到）：



推论 6.2 假设 p 为素数，则对任意的 $a \in QR_p$，恰好存在 a 模 p 的两个平方根。用 x 表示其中的一个，则另一个是 $-x\ (=p-x)$。 □



6.6.2 求模为合数时的平方根

由定理 6.8，对于 $n = pq$，其中 p, q 为素数，$\mathbb{Z}_n^*$ 同构于$\mathbb{Z}_p^* \times \mathbb{Z}_q^*$。由于同构关系保持运算，所以

$$x^2 \equiv y \pmod n$$

成立当且仅当对模为素数 p 和 q 时成立。因此，如果给出 n 的分解，用算法 6.5 可以求得模 n 的平方根。

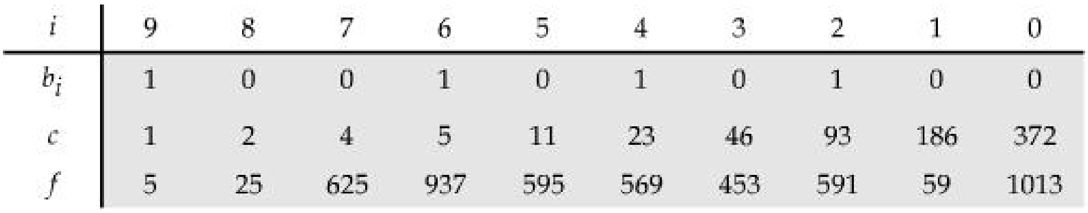
显然算法 6.5 的时间复杂度是 $O_B((\log n)^4)$。

由推论 6.2，$y \pmod p$ 有两个不同的平方根，分别记为 x_p 和 $p-x_p$；对于 $y \pmod q$ 同样也有两个，记为 x_q 和 $q-x_q$。由$\mathbb{Z}_n^*$ 和$\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ 之间的同构关系(定理 6.8)，我们知道 $y \in QR_n$ 在 $\mathbb{Z}_n^*$ 中有 4 个平方根。由算法 6.5，这 4 个根是

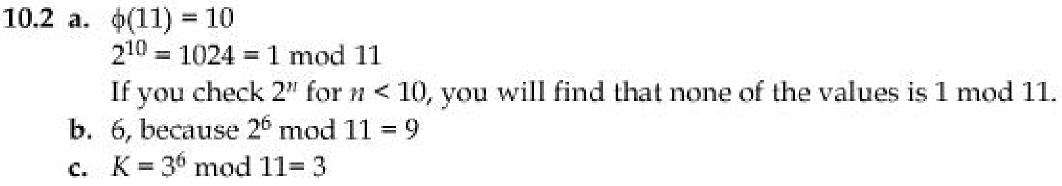
$$\left.\begin{array}{ll} x_1 \equiv \bar{1}_p x_p & + \bar{1}_q x_q \\ x_2 \equiv \bar{1}_p x_p & + \bar{1}_q (q - x_q) \\ x_3 \equiv \bar{1}_p (p - x_p) & + \bar{1}_q x_q \\ x_4 \equiv \bar{1}_p (p - x_p) & + \bar{1}_q (q - x_q) \end{array}\right\} \pmod n \qquad (6.6.8)$$

）

9.16



i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	1	0	1	0	1	0	0
c	1	2	4	5	11	23	46	93	186	372
f	5	25	625	937	595	569	453	591	59	1013



10.2 a. $\phi(11) = 10$

$2^{10} = 1024 = 1 \bmod 11$

If you check 2^n for $n < 10$, you will find that none of the values is 1 mod 11.

b. 6, because $2^6 \bmod 11 = 9$

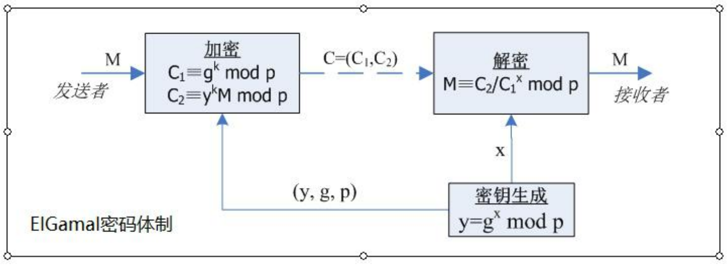
c. $K = 3^6 \bmod 11 = 3$

10.6

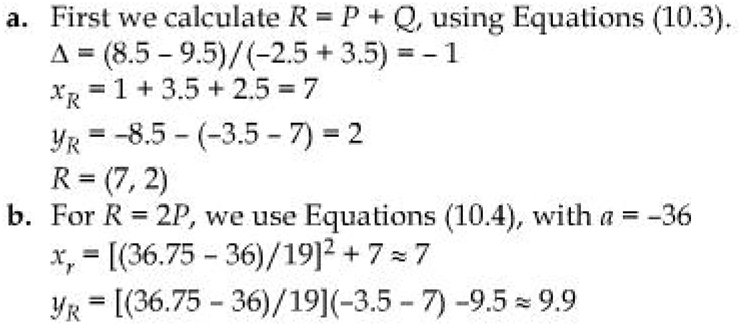
(1) $C_1 \equiv g^k \bmod p = 7^2 \bmod 71 = 49$， $C_2 \equiv y_B{}^k M \bmod p = (3^2 \times 30) \bmod 71 =$ 57 所以密文为 $C=(C_1, C_2)=(49, 57)$。

(2) 由 $7^k \bmod 71 = 59$ ，穷举 k 可得 k=3 。

所以 $C_2 = (3^k \times 30) \bmod 71 = (3^3 \times 30) \bmod 71 = 29$。



10.10



a. First we calculate $R = P + Q$, using Equations (10.3).

$\Delta = (8.5 - 9.5)/(-2.5 + 3.5) = -1$

$x_R = 1 + 3.5 + 2.5 = 7$

$y_R = -8.5 - (-3.5 - 7) = 2$

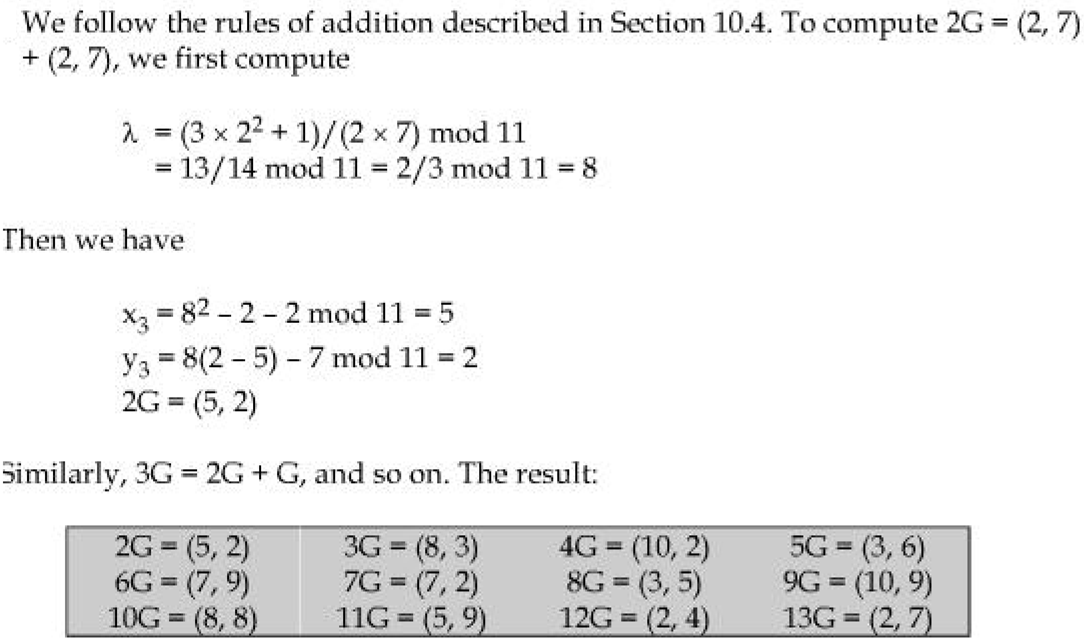
$R = (7, 2)$

b. For $R = 2P$, we use Equations (10.4), with $a = -36$

$x_r = [(36.75 - 36)/19]^2 + 7 \approx 7$

$y_R = [(36.75 - 36)/19](-3.5 - 7) - 9.5 \approx 9.9$

10.14



We follow the rules of addition described in Section 10.4. To compute 2G = (2, 7) + (2, 7), we first compute

$$\begin{aligned}\lambda &= (3 \times 2^2 + 1)/(2 \times 7) \bmod 11 \\ &= 13/14 \bmod 11 = 2/3 \bmod 11 = 8\end{aligned}$$

Then we have

$x_3 = 8^2 - 2 - 2 \bmod 11 = 5$

$y_3 = 8(2 - 5) - 7 \bmod 11 = 2$

2G = (5, 2)

Similarly, 3G = 2G + G, and so on. The result:

2G = (5, 2)	3G = (8, 3)	4G = (10, 2)	5G = (3, 6)
6G = (7, 9)	7G = (7, 2)	8G = (3, 5)	9G = (10, 9)
10G = (8, 8)	11G = (5, 9)	12G = (2, 4)	13G = (2, 7)

10.15

(a) B 的公钥 $P_B = n_B G = 7G = (7, 2)$

(b) $C_1 = kG = 3G = (8, 3)$, $C_2 = P_m + kP_B = (10,9) + 3(7, 2) = (10,9) + (3, 5) = (10, 2)$，所以密文 $C_m = \{C_1, C_2\} = \{ (8,3), (10, 2) \}$

(c) 解密过程为

$$C_2 - n_B C_1 = (P_m + kP_B) - n_B\,(kG) = (10, 2) - 7(8,3) = (10, 9) = P_m$$