

3DES: k1加密 -> k2解密 -> k3加密

DES 加密算法使用64位元的區塊大小和56位元的金鑰。在一個金鑰空間中，有256個可能的金鑰，3DES用三個金鑰，所以是256\*3 = 2168種組合

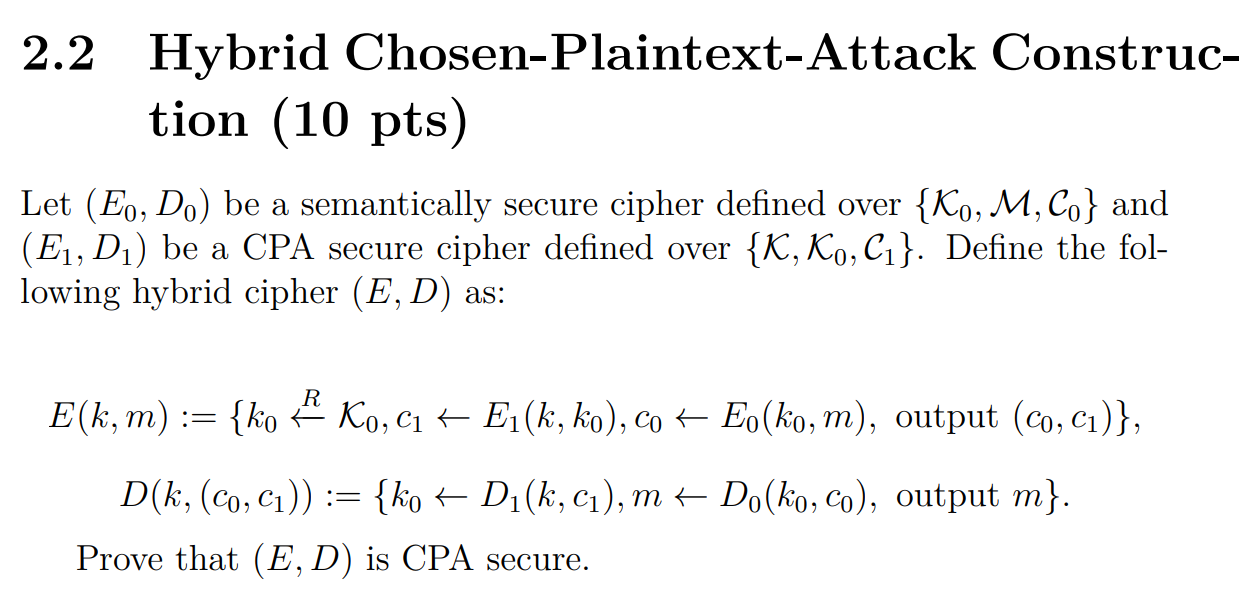
讓3DES等同於只用一把金鑰的可能情況:

情況1. 如果k1=k2，相當於只用k3加密

情況2. 如果k2=k3，相當於只用k1加密

如果k1, k2, k3是各自獨立選擇，情況1和情況2的機率就都是1/256\*2 = 1/2112

總體發生的機率是2/2112，小於1/288，所以可以忽略



CPA安全性要求對於任何給定的兩個m0和m1，如果攻擊者只能進行加密和解密操作，那麼他在區分兩個c0和c1的能力應該是可忽略的

用反證法證明(E,D)是CPA安全：

設(E,D)不安全(存在一個CPA攻擊者可以在可忽略的時間內區分兩個密文)

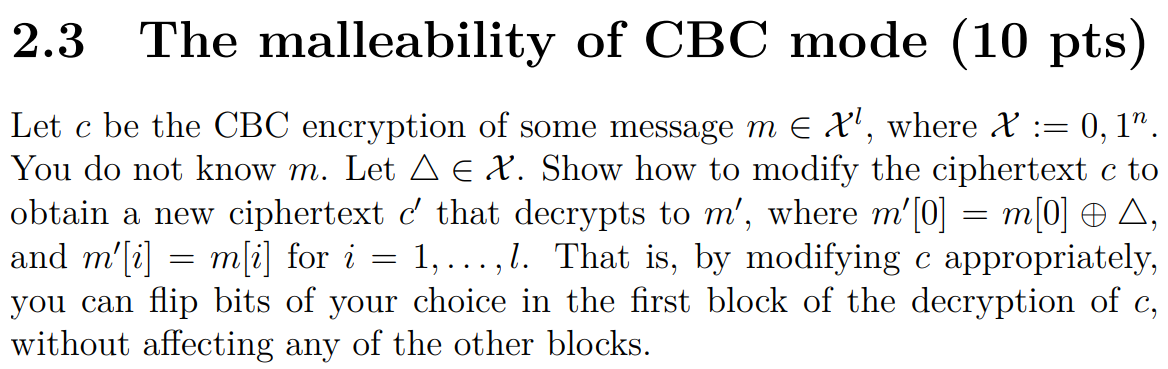
建構攻擊者A’，它是兩個子加密方案E0和E1的攻擊者

用A的結果區分(E,D)的密文 -> 違反(E0,D0)和(E1,D1)的安全性假設

由於A’和A的時間複雜度相同，且(E0,D0)和(E1,D1)是安全的

所以A’也無法區分(E,D)的密文

因此，如果(E0,D0)和(E1,D1)抗CPA攻擊，那(E,D)也是CPA安全

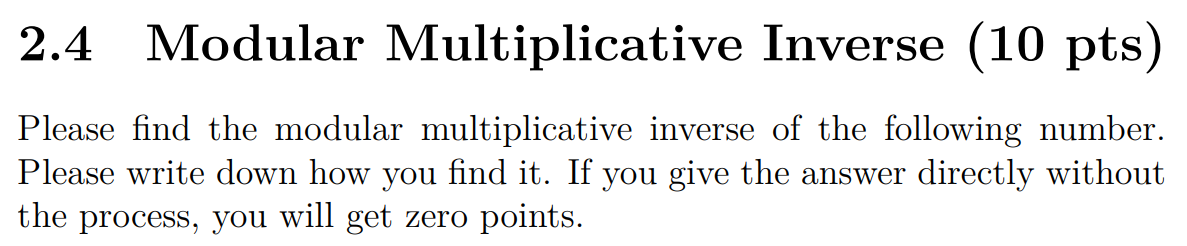


1. 解密c，得到m0, m1, m2,…,mn

2. 修改m0的位元，使m0[0] = m[0]⊕Δ

3. 將修改後的m0和原本的m1, m2,…,mn重新組合成新的m’

4. 將m’進行CBC加密，得到c’





**找到b使得 400b≡1(mod997)**

997 = 2\*400 + 197

400 = 2\*197 + 6

197 = 32\*6 + 5

6 = 1\*5 + 1

​**反回去**

1 = 6 − 1\*5

= 6 – 1\*(197 − 32\*6)

= 33\*6 – 1\*197

= 33\*(400 – 2\*197) – 1\*197

= 33\*400 – 67\*197

= 33\*400 – 67\*(997 − 2\*400)

= 167\*400 – 67\*997

​b = 167



**找到b使得 472b≡1(mod16651)**

16651 =35\*472+431

472 = 1\*431 + 41

431 = 10\*41 + 21

41 = 1\*21 + 20

21 = 1\*20 + 1

**反回去**

​1 = 21 – 1\*20

= 21 – 1\*(41 – 1\*21)

= 2\*21 – 1\*41

= 2\*(431 – 10\*41) – 1\*41

= 2\*431 – 21\*41

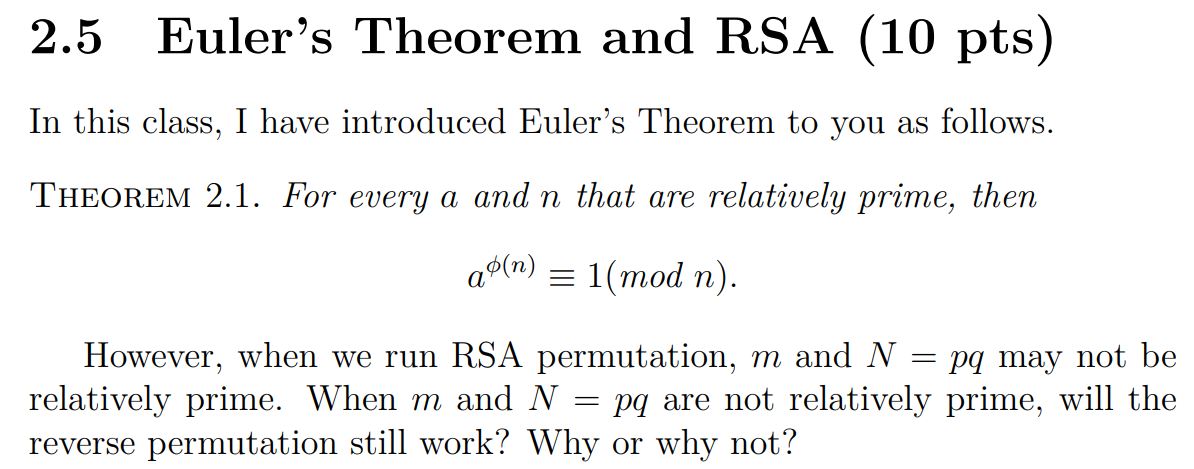
= 2\*431 – 21\*(472 – 1\*431)

= 23\*431 – 21\*472

= 23\*(16651 – 35\*472) – 21\*472

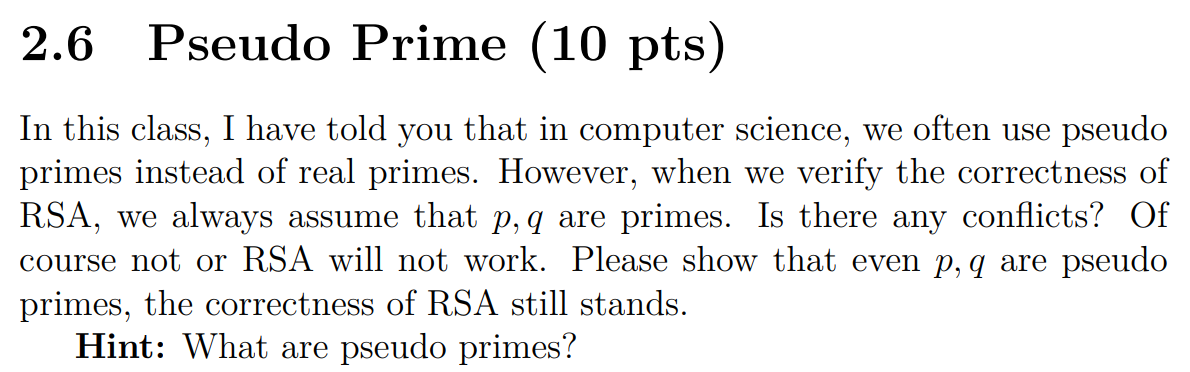
= −803\*472 + 23\*16651

​b = −803 但要找正的，所以是b = 16651−803 = 15848

​

在RSA加密中，我們選擇兩個大質數p和q，然後計算它們的乘積N = pq作為模數，當m和N不互質(m不是N的倍數)，Euler's Theorem不再適用

因為Euler's Theorem僅適用於互質的a和n，這樣才能確保 aϕ(n) ≡ 1(mod n)，當m和N不互質時，m就不會滿足Euler's Theorem的條件，因此反向置換不正確



因為RSA難解的原因是N = pq的p和q不好找，不是因為其他特殊的質數性質，所以只要p和q夠大，加上N的因數分解夠困難，就會讓RSA達到安全的效果。