《计算机安全与密码学》复习题

1. 信息安全（计算机安全）目标是什么？

答：机密性（confidentiality）：防止未经授权的信息泄漏

完整性（integrity）：防止未经授权的信息篡改

可用性（avialbility）：防止未经授权的信息和资源截留

抗抵赖性、不可否认性、问责性、可说明性、可审查性（accountability）：

真实性（authenticity）：验证用户身份

2. 理解计算安全性(即 one-time pad 的理论安全性)

使用与消息一样长且无重复的随机密钥来加密信息，即对每个明文每次采用不同的代换表不可攻破，因为任何明文和任何密文间的映射都是随机的，密钥只使用一次

3. 列出并简要定义基于攻击者所知道信息的密码分析攻击类型。

(1)、唯密文分析（攻击），密码分析者取得一个或多个用同一密钥加密的密文；

(2)、已知明文分析（攻击），除要破译的密文外，密码分析者还取得一些用同一密钥加密的密文对；

(3)、选择明文分析（攻击），密码分析者可取得他所选择的任何明文所对应的密文（不包括他要恢复的明文），这些密文对和要破译的密文是用同一密钥加密的；

(4)、选择密文分析（攻击），密码分析者可取得他所选择的任何密文所对应的明文（要破译的密文除外），这些密文和明文和要破译的密文是用同一解密密钥解密的，它主要应用于公钥密码体制。。

4. 传统密码算法的两种基本运算是什么？

代换和置换

前者是将明文中的每个元素映射成另外一个元素；后者是将明文中的元素重新排列。

5. 流密码和分组密码区别是什么？各有什么优缺点？

分组密码每次处理一个输入分组，对应输出一个分组；流密码是连续地处理输入元素，每次输出一个元素

流密码 Stream: 每次加密数据流的一位或者一个字节。连续处理输入分组，一次输出一个元素，速度较快

6. 已知密文 ILPQPUN 使用的是移位密码，试解密（提示：明文为有意义的英文）。

答：原文：　　ILPQPUN

移动 1 位：HKOPOTM　移动 2 位：GJNONSL 移动 3 位：FIMNMRK　移动 4 位：EHLMLQJ

移动 5 位：DGKLKPI　　移动 6 位：CFJKJOH 移动 7 位：BEIJING　　明文为 BEIJING。

7. 利用 playfair 密码加密明文 bookstore，密钥词是（HARPSICOD），所得的密文是什么？

I/JD RG LR QD HG

解答：生成 playfair 矩阵：

H	A	R	P	S
I/J	C	O	D	B
E	F	G	K	L
M	N	Q	T	U
V	W	X	Y	Z

根据矩阵加密为：

bo ok st or ex

I/JD DG PU GO GV

8. 用密钥 largest 构造一个 playfair 矩阵，并加密以下消息：

Must see you over Cadogan West. Coming at once.

注：该消息摘自 Sherlock Holmes 的故事 The Adventure of the Bruce-Partington Plans.

解答：矩阵为：

L	A	R	G	E
S	T	B	C	D
F	H	I/J	K	M
N	O	P	Q	U
V	W	X	Y	Z

加密为：UZTBDLGZPNNWLGTGTUEROVLDBDUHFPERHWQSRZ

9. 当海军上尉 John F.Kennedy 管理的美国巡逻船 PT-109 被日本毁灭者击沉时，位于澳大利亚的一个无线站截获了一条用 Playfair 密码加密的消息：

KXJEY UREBE ZWEHE WRYTU HEYFS

KREHE GOYFI WTTTU OLKSY CAJPO

BOTEI ZONTX BYBWT GONEY CUZWR

GDSON SXBOU YWRHE BAAHY USEDQ

密钥为 royal new Zealand navy.请解密这条消息，将 TT 换为 tt.

解答：PT BOAT ONE OWE NINE LOST IN ACTION IN BLACKETT STRAIT TWO MILES SW MERESU COVE X CREW OF TWELVE X REQUEST ANY INFORMATION

10. 用密钥词 cat 实现 vigenere 密码，加密明文 vigenere coper，所得的密文是什么？

解答：

Key: catcatca tcatcatcat

Plaintext: vigenere coper

Chipertext: XIZGNXTE VQPXT

11. 用 vigenere 密码加密单词 explanation.密钥为 leg .

解答：key: legleglegle

plaintext: explanation

ciphertext: PBVWETLXOZR

12. 假定有一个密钥 2431 的列置换密码，则明文 can you understand 的密文是多少？

YNSDCODTNURNAUEA

Key: 2 4 3 1

Plaintext: c a n y

o u u n

d e r s

t a n d

Chipertext: YNSDCODTNURNAUEA

13. 什么是乘积密码？

多步代换和置换，依次使用两个或两个以上的基本密码，所得结果的密码强度将强与所有单个密码的强度.

14. 混淆和扩散的区别是什么？

扩散（**Diffusion):**明文的统计结构被扩散消失到密文的,使得明文和密文之间的统计关系

尽量复杂.即让每个明文数字尽可能地影响多个密文数字

混淆**(confusion)**：使得密文的统计特性与密钥的取值之间的关系尽量复杂，阻止攻击者发现密钥

15. S-Box 的概念

S 盒用在 DES 算法中，每个 s 盒都由 6 位输入产生 4 位输出，所有说，s 盒定义了一个普通的可逆代换。相当程度上，DES 的强度取决于 s 盒的设计，但是，s 盒的构造方法是不公开的

16. 下表是DES 算法中S4 盒的选择矩阵，如果其输入为101011，则输出为

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

解、取输入首尾两位作为行号：11

取中间4位作为列号：0101

即第3行第5列：1

所以输出为四位二进制：0001

17. 这个问题给出了用一轮 DES 加密的具体数值的例子。我们假设明文和密钥 K 有相同的位模式，即

用十六进制表示为：0 1 2 3 4 5 6 7 8 9 A B C D E F

用二进制表示为： 0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

(a) 推导第一轮的子密钥 K_1

(b) 推导 L_0，R_0 。

(c) 扩展 R_0 得到 $E[R_0]$，其中 E[.]是表 3.2 的扩展函数。

(d) 计算 $A= E[R_0] \oplus K_1$

(e) 把(d) 的 48 位结果分成 6 位（数据）一组的集合并求对应 S 盒代替的值。

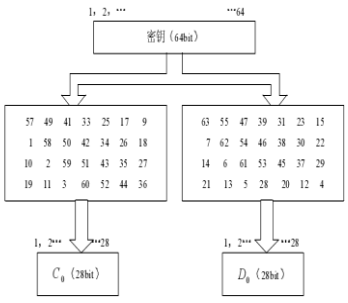
(f) 将(e)的结果连接起来获得一个 32 位的结果 B。

(g) 应用置换获得 P(B)。

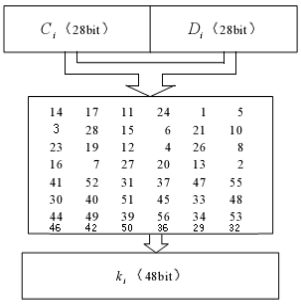
(h) 计算 $R_1= P(B) \oplus L_0$

(i) 写出密文。

子密钥生成



置换选择 1 PC-1



置换选择 2 PC-1

解答：**a.** **（根据上面 3 张图进行子密钥生成）**First, pass the 64-bit input through PC-1 to produce a 56-bit result. Then perform a left circular shift separately on the two 28-bit halves. Finally, pass the 56-bit result through PC-2 to produce the 48-bit *K*1.（首先根据 PC-1 将 64 位初始密钥转换为 56 位，然后将左右 28 位分别左循环移一位，最后，根据 PC-2 将 56 位置换选择为 48 位，即 K1 ）：

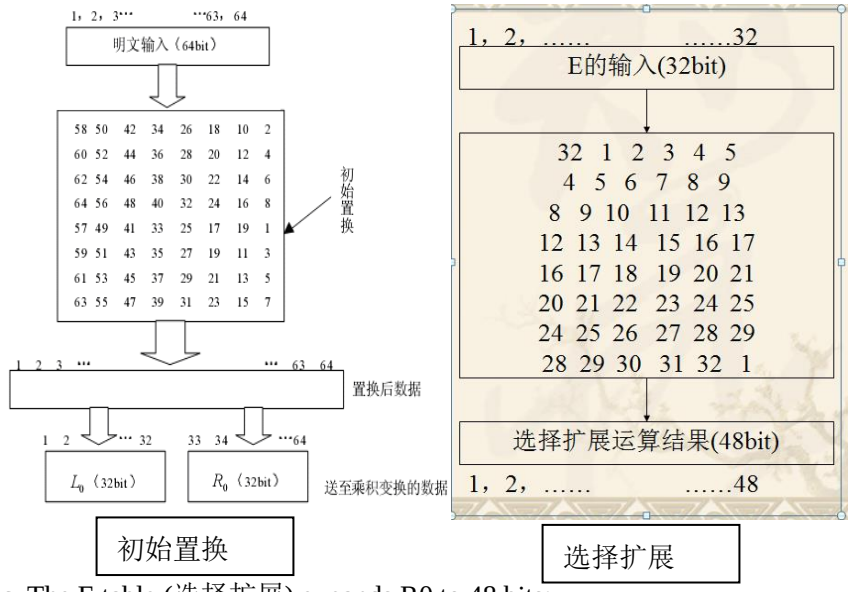
in binary notation: 0000 1011 0000 0010 0110 0111
1001 1011 0100 1001 1010 0101

in hexadecimal notation: 0 B 0 2 6 7 9 B 4 9 A 5

b. L0, R0 are derived by passing the 64-plaintext through IP (初始置换):

L0 = 1100 1100 0000 0000 1100 1100 1111 1111

R0 = 1111 0000 1010 1010 1111 0000 1010 1010



c. The E table (选择扩展) expands R0 to 48 bits:

E(R0) = 01110 100001 010101 010101 011110 100001 010101 010101

d. A = E[R_0] ⊕ K_1=011100 010001 011100 110010 111000 010101 110011 110000

e.

$S_1^{00}(1110) = S_1^0(14) = 0$ (base 10) = 0000 (base 2)

$S_2^{01}(1000) = S_2^1(8) = 12$ (base 10) = 1100 (base 2)

$S_3^{00}(1110) = S_3^0(14) = 2$ (base 10) = 0010 (base 2)

$S_4^{10}(1001) = S_4^2(9) = 1$ (base 10) = 0001 (base 2)

$S_5^{10}(1100) = S_5^2(12) = 6$ (base 10) = 0110 (base 2)

$S_6^{01}(1010) = S_6^1(10) = 13$ (base 10) = 1101 (base 2)

$S_7^{11}(1001) = S_7^3(9) = 5$ (base 10) = 0101 (base 2)

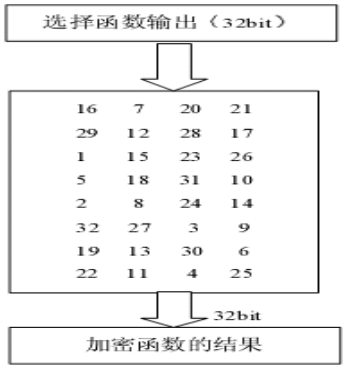
$S_8^{10}(1000) = S_8^2(8) = 0$ (base 10) = 0000 (base 2)

列 行	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	S_1
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	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	S_2
3	13	8	10	1	3	15	4	2	11	6	7	42	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	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	S_3
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	S_4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	S_5
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	S_6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	S_7
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	1	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	S_8
3	2	1	14	7	1	10	7	13	15	12	9	0	3	5	6	11	

S 盒

f. B = 0000 1100 0010 0001 0110 1101 0101 0000

g. 按照下图对 f 的 32 位结果进行变换, P(B) = 1001 0010 0001 1100 0010 0000 1001 1100



h. R1 = P(B) $\oplus L_0$= 0101 1110 0001 1100 1110 1100 0110 0011

i. L1 = R0. The ciphertext is the concatenation of L1 and R1.

18. AES 与 DES 相比有优点？3DES 与 DES 相比的变化有哪些？什么是 2DES 中的中

间相遇攻击?

(1) AES 更安全。

(2) 3DES 增加了 1 到 2 个密钥，进行多轮 DES，安全性更高。

(3) C = EK2(EK1(P)) ⇒ X = EK1(P) = DK2(C)

给定明文密文对(P,C)

对所有 256 个密钥,加密 P,对结果按 X 排序与 T 中

对所有 256 个密钥,解密 C,解密结果与 T 中的值比较

找出 K1,K2 使得 EK1(P) = DK2(C)

用 k1 和 k2 对 P 加密，若结果为 C，则认定这两个密钥为正确的密钥

19. 分组密码的工作模式有哪些？及优缺点？

A. ECB，电码本模式，一次处理 64 位明文，每次使用相同的密钥加密。任何 64 位的明文组都有唯一的密文与之对应，有“结构化”的缺点。

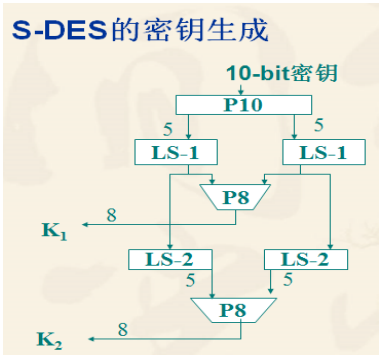
B. CBC，密码分组连接模式，克服了 ECB 中“结构化”的缺点，同样的明文变成密文之后就不同了，而且加密必须从头到尾

C. CFB，密码反馈模式．一次处理M位，上一个分组的密文产生一个伪随机数输出的加密算法的输入，该输出与明文的异或，作为下一个分组的输入。

D. OFB，输出反馈模式，与C F B基本相同，只是加密算法的输入是上一次 DES 的输出。

E. 计数器模式，计数器被初始化为某个值，并随着消息块的增加其值加 1，在于明文组异或得到密文组。也可用于流密码。

20. 下图为 S-DES 密钥生成图：



其中：

P10：3 5 2 7 4 10 1 9 8 6

P8：6 3 7 4 8 5 10 9

按照上述条件，若 K 选为(1010000010)，

产生的两个子密钥分别为：

K1=(1 0 1 0 0 1 0 0)

K2=(0 1 0 0 0 0 1 1)

21. 公钥和私钥的作用是什么？

用户的私钥是保密的，只知道给用户。用户的公共密钥提供给他人使用。可以用私钥加密，可以由任何人与公共密钥验证签名。或公共密钥可以用于加密信息，只能由拥有私钥解密。

22. 求 GCD（560，1547） 结果为 7

.5 $a = 560, b = 1547, ??? = \gcd\{560, 1547\}$

$1547 = 2 \times 560 + 427$

$427 = 1547 - (2 \times 560)$

$560 = 1 \times 427 + 133$

$133 = 560 - 427 = -1547 + (3 \times 560)$

$427 = 3 \times 133 + 28$

$28 = 427 - (3 \times 133) = (4 \times 1547) - (11 \times 560)$

$133 = 4 \times 28 + 21$

$21 = 133 - (4 \times 28) = (-17 \times 1547) + (47 \times 560)$

$28 = 1 \times 21 + 7$

$7 = 28 - 21 = (21 \times 1547) - (58 \times 560)$

$21 = 3 \times 7 + 0$

$7 = \gcd\{560, 1547\} = (21 \times 1547) - (58 \times 560), \quad x = -58, y = 21$

23. 求 GCD（645，1807） 结果为 1 ，此时可求出 645 在模 1807 下的乘法逆元。

$a = 654, b = 1807, ??? = \gcd\{645, 1807\}$

$1807 = 2 \times 654 + 499$

$499 = 1807 - (2 \times 654)$

$654 = 1 \times 499 + 155$

$155 = 654 - 499 = -1807 + (3 \times 654)$

$499 = 3 \times 155 + 34$

$34 = 499 - (3 \times 155) = (4 \times 1807) - (11 \times 654)$

$155 = 4 \times 34 + 19$

$19 = 155 - (4 \times 34) = (-17 \times 1807) + (47 \times 654)$

$34 = 1 \times 19 + 15$

$15 = 34 - 19 = (21 \times 1807) - (58 \times 654)$

$19 = 1 \times 15 + 4$

$4 = 19 - 15 = (-38 \times 1807) + (105 \times 654)$

接下图

$15 = 3 \times 4 + 3$

$3 = 15 - (3 \times 4) = (135 \times 1807) - (373 \times 654)$

$4 = 1 \times 3 + 1$

$1 = 4 - 3 = (-173 \times 1807) + (478 \times 654)$

$3 = 3 \times 1 + 0$

故

$1 = \gcd\{560, 1547\} = (-173 \times 1807) + (478 \times 654), \quad x = 478, y = 173$

$477 = 654^{-1} \pmod{1807} \Leftrightarrow 1 = (478 \times 654) \pmod{1807}$

由 $1 = (478 \times 654) \pmod{1807}$ 可知 478 和 654 在模 1807 下互为乘法逆元。

24. 求 GCD（123，277） 结果为 1。此时可求出 123 在模 277 下的乘法逆元

例 11.7

$a = 123, b = 277, ??? = \gcd\{123, 277\}$

$277 = 2 \times 123 + 31$

$31 = 277 - (2 \times 123)$

$123 = 3 \times 31 + 30$

$30 = 123 - (3 \times 31) = (7 \times 123) - (3 \times 277)$

$31 = 1 \times 30 + 1$

$1 = 31 - (1 \times 30) = (-9 \times 123) + (4 \times 277)$

$30 = 30 \times 1$

故

$1 = \gcd\{277, 123\} = (-9 \times 123) + (4 \times 277)$

$9 = 123^{-1} \pmod{277} \Leftrightarrow 1 = (9 \times 123) \pmod{277}$

由上述结果可知：9 和 123 在模 277 下互为乘法逆元。

25. 求 GCD（24140，16762）

gcd(24140, 16762) = gcd(16762, 7378) = gcd(7378, 2006) = gcd(2006, 1360) = gcd(1360, 646) = gcd (646, 68) = gcd(68, 34) = gcd(34, 0) = 34

26. 求 GCD（4655，12075）结果为 35

27. 欧拉函数 φ(n)表示所有小于正整数 n 且与 n 互素的正整数的个数。在表中给出了当 n=2-13 时欧拉函数的值：

n	2	3	4	5	6	7	8	9	10	11	12	13
φ(n)	1	2	2	4	2	6	4	6	4	10	4	12

28. 下面是 $a^b \bmod n$ 的快速指数算法。请注意，这里的变量 c 不是必需的，引入它只是为了便于解释算法，c 的终值既是指数。f 的终值即为算法的结果 。

```
c=0; f=1;
for i=k downto 0
  do c=c*2
     f=(f*f)mod n
if bi=1
   then c=c+1
        f=(f*a) mod n
return f
```

注：整数 b 表示为二进制 $b_k b_{k-1} b_{k-2} \ldots b_0$

表 计算 $a^b \bmod n$ 的快速模幂算法，其中，a=7,b=560=1000110000,n=561

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
f	7	49	157	526	160	241	298	166	67	1

运用上述算法，计算 $5^{596} \bmod 1234$.给出计算中的步骤。

解答：

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

29. RSA 算法中密钥的生成和加密解密过程。

（1）密钥生成过程：

① 选两个保密的大素数 p 和 q。

② 计算 n=p×q，φ(n)=(p-1)(q-1),其中 φ(n)是 n 的欧拉函数值。

③ 选一整数 e，满足 1<e<φ(n)，且 gcd(φ(n),e)=1。

④ 计算 d，满足 $d \cdot e \equiv 1 \bmod \varphi(n)$,即 d 是 e 在模 φ(n)下的乘法逆元，因 e 与 φ(n)互素，由模运算可知，它的乘法逆元一定存在。

⑤ 以{e,n}为公开钥,{d,n}为秘密钥。

（2）加密

加密时首先将明文比特串分组，使得每个分组对应的十进制数小于 n，即分组长度小于 $\log_2 n$。然后对每个明文分组 m，作加密运算： $c \equiv m^e \bmod n$

（3）解密

对密文分组的解密运算为：

$m \equiv c^d \bmod n$

30. RSA 算法计算实例（给定 p, q, e, m/c，计算 n, $\varphi(n)$, d, c/m）

1. 选择素数：p=17 & q=11
2. 计算 $n = pq$ =17×11=187
3. 计算 $\varphi(n)=(p-1)(q-1)$=16×10=160
4. 选择 e ： gcd(e,160)=1；选择 e=7
5. 确定 d： de=1 mod 160 and $d < 160$， d=23

因为 23×7=161= 1×160+1

6. 公钥 PK={7,187}
7. 私钥 SK={23,187}
8. RSA 的加解密为:

给定消息 M = 88 (88<187)

加密:

$C = 88^7 \bmod 187 = 11$

解密:

$M = 11^{23} \bmod 187 = 88$

31. 用 RSA 算法对下列数据实现加密和解密：

(a) p=3;q=11;e=7;M=5

(b) p=5;q=11;e=3;M=9

(c) p=7;q=11;e=17;M=8

(d) p=11;q=13;e=11;M=7

(e) p=17;q=31;e=7;M=2

解答：**a.** n = 33; $\phi(n)$ = 20; d = 3; C = 26.

b. n = 55; $\phi(n)$ = 40; d = 27; C = 14.

c. n = 77; $\phi(n)$ = 60; d = 53; C = 57.

d. n = 143; $\phi(n)$ = 120; d = 11; C = 106.

e. n = 527; $\phi(n)$ = 480; d = 343; C = 128. For decryption, we have

$$128^{343} \bmod 527 = 128^{256} \times 128^{64} \times 128^{16} \times 128^{4} \times 128^{2} \times 128^{1} \bmod 527$$
$$= 35 \times 256 \times 35 \times 101 \times 47 \times 128 = 2 \bmod 527$$
$$= 2 \bmod 257$$

32. 在使用 RSA 的公钥密码体制中，已截获发给某用户的密文 C=10，该用户的公钥 e=5,n=35,那么明文 M 是多少？

结果：5

33. 在 RSA 体制中，某给定用户的公钥 e=31,n=3599,那么该用户的私钥等于多少？提示：首先用试探法决定 p 和 q ，然后用扩展 Euclid 算法寻找 31mod$\phi(n)$的乘法逆元。

解答：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.

34. 对比对称算法和公钥算法？（建议从用途，速度和效率等方面）

对称算法：速度快，主要用于数据加密，只有一个密钥。

公钥算法：速度较慢，主要用于数字签名和密钥交换，有两个密钥

35. 消息认证码的概念和基本用途？

MAC（Message Authentication Code），消息认证码，也是一种认证技术，它利用密钥来产生一个固定长度的短数据块，并将数据块附加在消息之后，格式如：MAC（M）|| M。消息和 MAC 一起发送到接受方。从而，接受方可以知道消息没有经过篡改,真正来自发送方（MAC 的密钥）和消息的时效性（如果 MAC 中包含序列号）。从这个层面来说，hash 是没有密钥的 MAC

36. 散列函数的基本用途有哪些？

保密、认证和签名

37. 消息认证码和散列函数有哪些区别？

散列函数(Hash)：将任意长度的消息变换为定长的消息摘要，并加以认证。

消息认证码(MAC)：依赖公开的函数（密钥控制下）对消息进行处理，生成定长的认证标识，并加以认证。

38. 什么是数字签名？如何理解 RSA 私钥运算结果做为数字签名?

【提示：最简单的数字签名是：$E_{SK(a)}$（ M） 即用 a 的私钥（SKa）加密消息 M，接受方 b 用 a 的公钥解密，得到 M，b 就可以认为 M 来自 a，因为其他人不可能有 a 的私钥；而且消息没有经过修改，因为修改后的秘文不能用 a 的公钥解开，从而实现了数字签名。】

39. 如何实现用签名进行身份和消息认证？【提示：上面算法的改进算法就可以实现用签名进行身份和报文鉴别：E_{SKa}（ H（M））||M 。先将消息 M 用 hash 算法（MD5 or SHA1）算出 mac（消息认证码），然后，用 a 的私钥加密此认证码，最后和原始的消息并在一起，发送到接受方 b。b 首先用 a 的公钥 PKa 解密前面部分，然后用同样的 hash 算法对 M 进行 hash 操作，比较两个结果是否相等。从而实现身份和消息认证。

40. 数字签名的作用是什么

当通信双方发生了下列情况时，数字签名技术必须能够解决引发的争端：

(1) 否认，发送方不承认自己发送过某一报文。

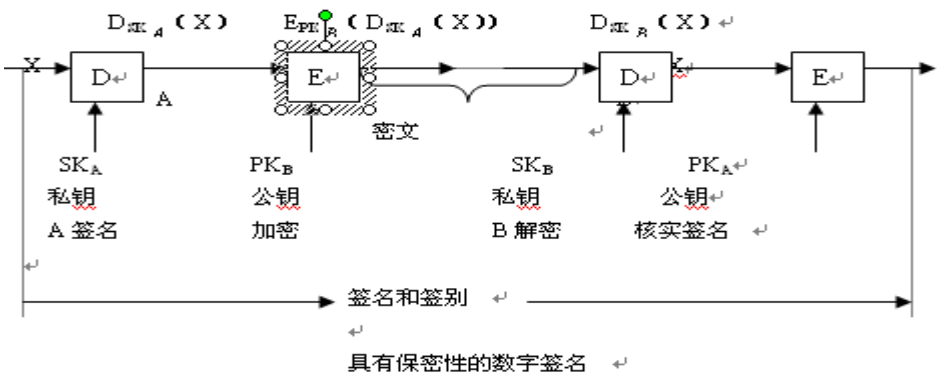
(2) 伪造，接收方自己伪造一份报文，并声称它来自发送方。

(3) 冒充，网络上的某个用户冒充另一个用户接收或发送报文。

(4) 篡改，接收方对收到的信息进行篡改。

41. 请用公开密钥密码体制描述具有保密性的数字签名。（可以用图示说明表示）

2、可以用图示说明表示为：



具有保密性的数字签名

42. 实体认证（身份认证）和消息认证的区别是什么？

身份认证是验证主体的真实身份与其所声称的身份是否符合的过程。消息认证是是一个证实收到的消息来自可信的源点且未被篡改的过程。即验证收到的消息确实是来自真正的发送方且未被修改的消息，也验证消息的顺序和及时性。是为了确认被认证的实体与一些特定数据项有着静态的联系，而身份认证主要是在连接建立或者在数据传送阶段的某些时刻使用的。

43. 什么是消息重放？有哪些方法可以抵御消息的重放攻击，各有什么特点？

消息重放：攻击者发送一个目的主机已接收过的包，来达到欺骗系统的目的。

对付重放攻击的一种方法是在认证交换中使用一个序列号来给每一个消息报文编号。仅当收到的消息序数顺序合法时才接受之。但这种方法的困难是要求双方必须保持上次消息的序号。

两种更为一般的方法是：

时间戳：A 接受一个新消息仅当该消息包含一个时间戳，该时间戳在 A 看来，是足够接近 A 所知道的当前时间；这种方法要求不同参与者之间的时钟需要同步。

挑战/应答方式。（Challenge/Response）A 期望从 B 获得一个新消息，首先发给 B 一个临时值(challenge)，并要求后续从 B 收到的消息（response）包含正确的这个临时值。

挑战问/应答方法不适应非连接性的应用，因为它要求在传输开始之前先有握手的额外开销，这就抵消了无连接通信的主要特点。