

---

## CHAPTER 10

### *Symmetric-Key Cryptography*

(Solution to Practice Set)

#### Review Questions

1. Symmetric-key cryptography is based on sharing secrecy; asymmetric-key cryptography is based on personal secrecy.
  2. In asymmetric-key cryptography, each entity has a pair of public/private key. The public key is universal; the private key is personal. In symmetric-key cryptography a shared secret key is used for secret communication between two entities.
  3. In cryptography, a trapdoor is a secret with which Bob can use a feasible algorithm to decrypt the ciphertext. If Eve does not know the trapdoor, she needs to use an algorithm which is normally infeasible.
  4. In the knapsack cryptosystem, if we are told which elements, from a predefined set of numbers, are in a knapsack, we can easily calculate the sum of the numbers; if we are told the sum, it is difficult to say which elements are in the knapsack.
    - a. The one-way function is the multiplication of the row matrix  $\mathbf{x}$  by the column matrix  $\mathbf{a}$  to get  $s = \mathbf{x} \times \mathbf{a}$ . Given  $\mathbf{x}$  and  $\mathbf{a}$ , it is easy to calculate  $s$ ; given  $s$  and  $\mathbf{a}$ , it is difficult to calculate  $\mathbf{x}$ .
    - b. The trapdoor in this system is the value of  $r$  and  $n$  that allow Bob to calculate the value of  $s' = r^{-1} \times s$ . In the matrix multiplication  $s' = \mathbf{x} \times \mathbf{b}$ , the matrix  $\mathbf{b}$  can be calculated if it is a superincreasing column matrix (tuple).
    - c. The public key is the  $k$ -tuple  $\mathbf{a}$ . The private key is  $n, r$ ; and the  $k$ -tuple  $\mathbf{b}$ .
    - d. The security of the system depends on the size of the two matrices  $\mathbf{x}$  and  $\mathbf{a}$ . If they are very large, it is difficult to find  $\mathbf{x}$  given  $s$  and  $\mathbf{a}$ . However, today, knapsack cryptosystem is not considered secure; it is broken.
- 
5. RSA uses two exponents,  $e$  and  $d$ , where  $e$  is public and  $d$  is private. Alice calculates  $C = P^e \bmod n$  to create ciphertext  $C$  from plaintext  $P$ ; Bob uses  $P = C^d \bmod n$  to retrieve the plaintext sent by Alice.
    - a. The one-way function is the  $C = P^e \bmod n$ . Given  $P$  and  $e$ , it is easy to calculate  $C$ ; given  $C$  and  $e$ , it is difficult to calculate  $P$  if  $n$  is very large.

- b.** The trapdoor in this system is the value of  $d$ , which enables Bob to use  $P = C^d \bmod n$ .
  - c.** The public key is the tuple  $(e, n)$ . The private key is  $d$ .
  - d.** The security of RSA mainly depend on the factorization of  $n$ . If  $n$  is very large, and the value of  $e$  and  $d$  are chosen properly, the system is secure.
- 6.** The Rabin cryptosystem is a variation of the RSA cryptosystem. RSA is based on the exponentiation congruence; Rabin is based on quadratic congruence. The Rabin cryptosystem can be thought of as an RSA cryptosystem in which  $e = 2$  and  $d = 1/2$ .
- a.** The one way function is  $C = P^2 \bmod n$ . Given  $P$ , it is easy to calculate  $C$ ; given  $C$ , it difficult to calculate  $P$  if  $n$  is very large.
  - b.** The trapdoor is the factorization of  $n = p \times q$ . Bob knows the value of  $p$  and  $q$ , but Eve does not. Bob uses the Chinese remainder theorem and the values of  $p$  and  $q$ , to find  $P$ .
  - c.** The public key is  $n$ . The private key is the tuple  $(p$  and  $q)$ .
  - d.** Security of Rabin cryptosystem is basically depends on how it difficult to factor  $n$ .
- 7.** ElGamal is based on discrete logarithm problem. The plaintext is masked with  $e_1^{nd}$  to create the ciphertext. Part of the mask is created by Bob and become public; the other part is created by Alice.
- a.** The one-way function is  $C = \text{mask}(P)$ . Given  $P$  and the mask, it is easy to calculate the  $C$ ; given  $C$  is difficult to unmask  $P$ .
  - b.** The trapdoor is the value of  $d$  that enables Bob to unmask  $C$ .
  - c.** The public key is  $(e_1, e_2$  and  $n)$ . The private key is  $d$ .
  - d.** The security of ElGamal depends on two points;  $p$  should be very large and Alice needs to select a new  $r$  for each encryption.
- 8.** ECC is based on elliptic curves. We can defined a group  $GF(p)$  or  $GF(2^n)$  in which the elements are points on an elliptic curve. ECC simulates the idea of ElGamal using the above-mentioned groups.
- a.** The one-way function in ECC is the idea of multiplying an integer by a point to get a new point on the curve. If the original point is given, the new point can be calculated with polynomial complexity. If the new point is given, it is very hard to calculate the original point without knowing the trapdoor.
  - b.** The trapdoor is the value of  $d$  that enables Bob to calculate the original point on the curve using an algorithm with polynomial complexity.
  - c.** The public key is  $(e_1, e_2,$  and  $E)$  in which  $e_1$  and  $e_2$  are two points on the curve  $E$ . The private key is  $d$ .
  - d.** The security of ECC is based on the difficulty of solving elliptic curve logarithms.

9. An elliptic curve is an equation in two variables similar to the equations used to calculate the length of a curve in the circumference of an ellipse. Elliptic curves with points belonging to the groups  $GF(p)$  or  $GF(2^n)$  are used in cryptography.
10. Operations are basically adding two points on the curve to get a new point on the curve.

## Exercises

11. We use the 7-bit representation of character "a" as the plaintext.

### Key Generation:

$r = [287, 451, 943, 762, 564, 86, 623] \rightarrow a = [623, 86, 564, 287, 451, 943, 762]$

### Encryption:

Plaintext:  $x = [1, 1, 0, 0, 0, 0, 1] \rightarrow$  Ciphertext:  $s = 1471$

### Decryption:

$$s' = (41^{-1}, 1471) \bmod 1001 = 293 \times 1471 \bmod 1001 = 573$$

$$x' = [0, 0, 0, 1, 0, 1, 1] \rightarrow \text{Plaintext } x = [1, 1, 0, 0, 0, 0, 1]$$

12.

- a.  $\phi(n) = \phi(13 \times 17) = 192 \rightarrow d = e^{-1} \bmod \phi(n) = 5^{-1} \bmod 192 = 77$
- b.  $\phi(n) = \phi(31 \times 127) = 3780 \rightarrow d = e^{-1} \bmod \phi(n) = 17^{-1} \bmod 3780 = 3113$
- c.  $n = p \times q = 437 \rightarrow \phi(n) = 396$ . **Since  $\gcd(e, \phi(n)) \neq 1$ , we cannot calculate  $d$  for this problem** This shows that we need to choose the value of  $e$  carefully. In this case  $\phi(n) = 396 = 2^2 \times 3^2 \times 11$ , which means  $e$  cannot be 2, 3, or 11. We can have  $e = 5$ , which means  $d = 5^{-1} \bmod 396 = 317$ .

13.  $n = 187 = 17 \times 11 \rightarrow \phi(n) = 17 \times 11 = 160 \rightarrow d = e^{-1} \bmod \phi(n) = 113$ . This proves that the value of  $n$  in RSA must be very large. We could find  $d$  because we could factor  $n$ . The modulus must be large enough to make the factorization infeasible.

14. We can create two equations of two unknowns ( $p$  and  $q$ ):

$$\begin{aligned} p \times q &= n \\ (p-1) \times (q-1) &= \phi(n) \end{aligned}$$

If we let  $A = (n - \phi(n) + 1)$ , then  $p^2 - A \times p + n = 0$ . This means

$$p = [A + (A^2 - 4 \times n)^{1/2}] / 2 \quad \text{and} \quad q = [A - (A^2 - 4 \times n)^{1/2}] / 2$$

For example, assume  $n = 209$  and  $\phi(n) = 180$ . Then  $A = 209 - 180 + 1 = 30$ . We have  $p = 19$  and  $q = 11$ .

**15.** In a real situation, the value of  $n$  is so large that it is impossible for Alice to check if  $n$  and  $e$  are chosen properly. In this toy problem, it is easy to see that  $n$  is not properly selected because  $n = 100$  cannot be factored into two primes ( $n = p \times q$ ). Although we can still encrypt the message using  $e = 13$  and  $n = 100$ , the encrypted message cannot be decrypted. The problem prove that Bob needs to first select  $p$  and  $q$  and be sure that they are primes before calculating  $n = p \times q$ . After the correct selection of  $n$ , then  $e$  needs to be selected in such a way that  $\phi(n)$  and  $e$  be coprimes. **This problem has no solution.**

**16.** Although in real life Alice cannot check to see if  $n$  and  $e$  are properly selected by Bob, in this toy problem she can. The value of  $n$  is proper because  $n = 107 \times 113$  (two primes). The value of  $e$  is also proper because  $e$ , which is 13 and  $\phi(n)$ , which is 11872, are coprime.

**a.** The plaintext is: **19070818260818261914200607**

**b.** For encryption, we create 4-digit blocks so that the value of each block be less than  $n$ .

|                             |          |                                                    |                |
|-----------------------------|----------|----------------------------------------------------|----------------|
| <b>P<sub>1</sub> = 1907</b> | <b>→</b> | <b>C<sub>1</sub> = 1907<sup>13</sup> mod 12091</b> | <b>= 10614</b> |
| <b>P<sub>2</sub> = 0818</b> | <b>→</b> | <b>C<sub>2</sub> = 0818<sup>13</sup> mod 12091</b> | <b>= 7787</b>  |
| <b>P<sub>3</sub> = 2608</b> | <b>→</b> | <b>C<sub>3</sub> = 2608<sup>13</sup> mod 12091</b> | <b>= 1618</b>  |
| <b>P<sub>4</sub> = 1826</b> | <b>→</b> | <b>C<sub>4</sub> = 1826<sup>13</sup> mod 12091</b> | <b>= 10717</b> |
| <b>P<sub>5</sub> = 1914</b> | <b>→</b> | <b>C<sub>5</sub> = 1914<sup>13</sup> mod 12091</b> | <b>= 4084</b>  |
| <b>P<sub>6</sub> = 2006</b> | <b>→</b> | <b>C<sub>6</sub> = 2006<sup>13</sup> mod 12091</b> | <b>= 6558</b>  |
| <b>P<sub>7</sub> = 07</b>   | <b>→</b> | <b>C<sub>7</sub> = 07<sup>13</sup> mod 12091</b>   | <b>= 6651</b>  |

**c.** In this toy problem, we can easily find  $d = e^{-1} \text{ mod } \phi(n) = 3653$ . Each block can be decrypted as

|                              |          |                                                       |               |
|------------------------------|----------|-------------------------------------------------------|---------------|
| <b>C<sub>1</sub> = 10614</b> | <b>→</b> | <b>P<sub>1</sub> = 10614<sup>3653</sup> mod 12091</b> | <b>= 1907</b> |
| <b>C<sub>2</sub> = 7787</b>  | <b>→</b> | <b>P<sub>2</sub> = 7787<sup>3653</sup> mod 12091</b>  | <b>= 0818</b> |
| <b>C<sub>3</sub> = 1618</b>  | <b>→</b> | <b>P<sub>3</sub> = 1618<sup>3653</sup> mod 12091</b>  | <b>= 2608</b> |
| <b>C<sub>4</sub> = 10717</b> | <b>→</b> | <b>P<sub>4</sub> = 10717<sup>3653</sup> mod 12091</b> | <b>= 1826</b> |
| <b>C<sub>5</sub> = 4084</b>  | <b>→</b> | <b>P<sub>5</sub> = 4084<sup>3653</sup> mod 12091</b>  | <b>= 1914</b> |
| <b>C<sub>6</sub> = 6558</b>  | <b>→</b> | <b>P<sub>6</sub> = 6558<sup>3653</sup> mod 12091</b>  | <b>= 2006</b> |
| <b>C<sub>7</sub> = 6651</b>  | <b>→</b> | <b>P<sub>7</sub> = 6651<sup>3653</sup> mod 12091</b>  | <b>= 07</b>   |

The plaintext is: 19070818**26**0818**26**1914200607 or "THIS IS TOUGH".

**17.**

**a.** If  $e = 1$ , there is no encryption:  $C = P$ . If Eve intercepts the ciphertext, she has the plaintext.

b. If  $e = 2$ , the cipher is actually Rabin, not RSA.

18. Eve can use a type of plaintext attack called short-message attack if she knows that each character is encrypted separately. Eve can write a simple program using  $e$  and  $n$  to find all plaintext/ciphertext character as shown in the following table.

|              |           |           |                  |                 |           |          |
|--------------|-----------|-----------|------------------|-----------------|-----------|----------|
| P = 0        | P = 1     | P = 2     | P = 3            | P = 4           | P = 5     | P = 6    |
| <b>C = 0</b> | C = 1     | C = 13958 | C = 2459         | <b>C = 6625</b> | C = 7340  | C = 8320 |
| P = 7        | P = 8     | P = 9     | P = 10           | P = 11          | P = 12    | P = 13   |
| C = 8911     | C = 10247 | C = 15310 | C = 16008        | C = 18021       | C = 12029 | C = 2232 |
| P = 14       | P = 15    | P = 16    | P = 17           | P = 18          | P = 19    | P = 20   |
| C = 4670     | C = 13504 | C = 11913 | C = 10706        | <b>C = 2968</b> | C = 8539  | C = 5671 |
| P = 21       | P = 22    | P = 23    | P = 24           | P = 25          |           |          |
| C = 11831    | C = 15284 | C = 4718  | <b>C = 17863</b> | C = 1160        |           |          |

Eve knows that the plaintext is made of P = 4, P = 0, P = 18, and P = 24. The plaintext is "easy". Eve can break the code because the set of plaintext values is small. Eve can try all of them using a program. The set should be very large. Even if Alice has short pieces of messages to send, she need to pad them to make the set large (see OAEP encryption/decryption).

19. Eve has intercepted  $C = 57$
- Eve chooses  $X = 17$  (which is in  $\mathbf{Z}_{143}^*$ ).
  - Eve calculates  $Y = C \times 17^7 \bmod 143 = 57 \times 17^7 \bmod 143 = 137$
  - Eve sends 137 to Bob and ask to decrypt it. The response is  $Z = 136$  (We know how to calculate this because we can easily find  $d = 103$ , but we assume that Eve cannot; she needs to send the ciphertext to Bob and receive the plaintext.
  - $P = (Z \times X^{-1}) \bmod 143 = (136 \times 17^{-1}) \bmod 143 = 8 \bmod 143$ . Which is the plaintext. This shows that RSA is very vulnerable to chosen-ciphertext attack.

20.

Intercepted ciphertext is  $C = 22$ .

$$C_1 = C^e \bmod n = 22^3 \bmod 35 = 8.$$

$$C_2 = C_1^e \bmod n = 8^3 \bmod 35 = 22.$$

$$\text{Since } C_2 = C = 22, P = C_1 = 8.$$

21. One solution is to use different padding with each plaintext. As we discussed in the text, using OAEP encryption, each time with different random  $r$ , removes the relationships between different plaintexts that are sent by Alice to Bob.

22. We first find  $n = p \times q = 47 \times 11 = 517$ .

$$\text{a. } C = P^2 \bmod 517 = 17^2 \bmod 517 = 289.$$

b.

$$a_1 = +C^{(p+1)/4} \bmod p = +289^{12} \bmod 47 = +17 \text{ and } a_2 = -17.$$

$$b_1 = +C^{(q+1)/4} \bmod q = +289^3 \bmod 11 = +5 \text{ and } b_2 = -5.$$

We have the following four sets, each of two equations to solve:

|                            |                           |               |             |
|----------------------------|---------------------------|---------------|-------------|
| $P_1 \equiv +17 \pmod{47}$ | $P_1 \equiv +5 \pmod{11}$ | $\rightarrow$ | $P_1 = 346$ |
| $P_2 \equiv +17 \pmod{47}$ | $P_2 \equiv -5 \pmod{11}$ | $\rightarrow$ | $P_2 = 17$  |
| $P_3 \equiv -17 \pmod{47}$ | $P_3 \equiv +5 \pmod{11}$ | $\rightarrow$ | $P_3 = 500$ |
| $P_4 \equiv -17 \pmod{47}$ | $P_4 \equiv -5 \pmod{11}$ | $\rightarrow$ | $P_4 = 171$ |

The only acceptable answer is  $P_2 = 17$ .

23.

- a. We choose  $e_1 = 3$  (a primitive root of  $p = 31$ ) and  $d = 10$ . Then we have  $e_2 = 3^{10} \pmod{31} = 25$ .
- b. The common factor for calculation of  $C_2$ 's is  $e_2^7 \pmod{31} = 25^7 \pmod{31} = 25$ .

|                       |                            |                                     |               |                |
|-----------------------|----------------------------|-------------------------------------|---------------|----------------|
| $P = \text{"H"} = 07$ | $C_1 = 3^7 \pmod{31} = 17$ | $C_2 = 07 \times 25 \pmod{31} = 20$ | $\rightarrow$ | $C = (17, 20)$ |
| $P = \text{"E"} = 04$ | $C_1 = 3^7 \pmod{31} = 17$ | $C_2 = 04 \times 25 \pmod{31} = 07$ | $\rightarrow$ | $C = (17, 07)$ |
| $P = \text{"L"} = 11$ | $C_1 = 3^7 \pmod{31} = 17$ | $C_2 = 11 \times 25 \pmod{31} = 27$ | $\rightarrow$ | $C = (17, 27)$ |
| $P = \text{"L"} = 11$ | $C_1 = 3^7 \pmod{31} = 17$ | $C_2 = 11 \times 25 \pmod{31} = 27$ | $\rightarrow$ | $C = (17, 27)$ |
| $P = \text{"O"} = 14$ | $C_1 = 3^7 \pmod{31} = 17$ | $C_2 = 14 \times 25 \pmod{31} = 09$ | $\rightarrow$ | $C = (17, 09)$ |

c.

|                |               |                                               |               |     |
|----------------|---------------|-----------------------------------------------|---------------|-----|
| $C = (17, 20)$ | $\rightarrow$ | $P = 20 \times (17^{10})^{-1} \pmod{31} = 07$ | $\rightarrow$ | "H" |
| $C = (17, 07)$ | $\rightarrow$ | $P = 07 \times (17^{10})^{-1} \pmod{31} = 04$ | $\rightarrow$ | "E" |
| $C = (17, 27)$ | $\rightarrow$ | $P = 27 \times (17^{10})^{-1} \pmod{31} = 11$ | $\rightarrow$ | "L" |
| $C = (17, 27)$ | $\rightarrow$ | $P = 27 \times (17^{10})^{-1} \pmod{31} = 11$ | $\rightarrow$ | "L" |
| $C = (17, 09)$ | $\rightarrow$ | $P = 09 \times (17^{10})^{-1} \pmod{31} = 14$ | $\rightarrow$ | "O" |

24. In general, the numeric value of  $C_1$  and  $C_2$  are different in a ciphertext. If these two values are swapped during transmission, Bob cannot correctly decrypt the ciphertext to get the plaintext because

$$C_2 \times (C_1^d)^{-1} \neq C_1 \times (C_2^d)^{-1}$$

25. Although the value of  $p$  (the modulus) and  $d$  are not given, we can assume a modulus which is greater than 17 and 37 and its primitive root is 2. We have chosen  $p = 53$  and  $d = 3$ . In this case, we have  $p = 53$ ,  $d = 3$ ,  $e_1 = 2$ , and  $e_2 = e_1^3 \pmod{53} = 8$ .
- a. Alice uses  $r = 9$  to encrypt two messages, 17 and 37. The values of ciphertexts are:  $C_1 = 35$ ,  $C_2 = 19$ ,  $C_1' = 35$  and  $C_2' = 32$ .
- b. Eve intercepts  $C_1 = 35$ ,  $C_2 = 19$ ,  $C_1' = 35$  and  $C_2' = 32$  and she knows  $P = 17$ . Eve can use the known-plaintext attack to find  $P'$ .

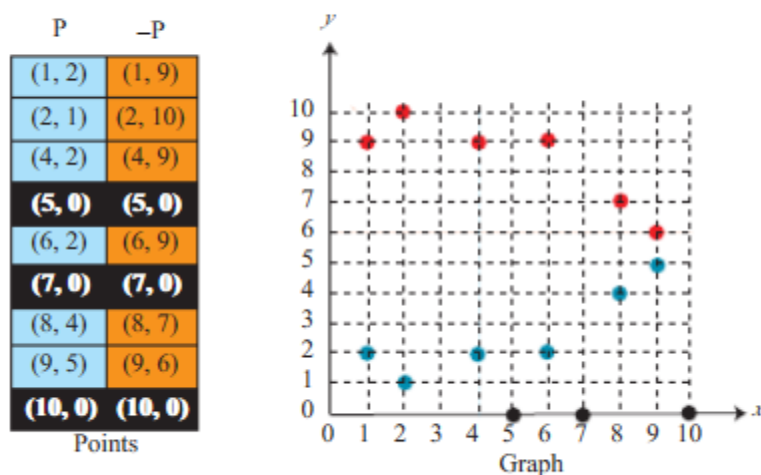
$$P' = C_2' \times (e_2')^{-1} \bmod p = C_2' \times (C_2 \times P^{-1})^{-1} \bmod p = C_2' \times C_2^{-1} \times P \bmod p$$

$$P' = 32 \times 19^{-1} \times 17 \bmod 53 = 32 \times 14 \times 17 \bmod 53 = 37$$

26.

- a.  $E(1, 2)$  means that  $a = 1$  and  $b = 2$  in the equation  $y^2 = x^3 + ax + b$ . The equation of the curve is then  $y^2 = x^3 + x + 2$ .
- b. Figure S10.26 shows the points on the curve.

**Figure S10.26** Part of solution to Exercise 26



- c. Bob chooses  $e_1 = (2, 1)$  and  $d = 3$ , then  $e_2 = 3 \times (2, 1) = (4, 9)$ . Public key is  $E_{11}(1, 2)$ ,  $e_1$ , and  $e_2$ . The private key is  $d$ .
- d. Assume Alice has the plaintext  $P = (4, 2)$  to send to Bob.
- e. Alice choose  $r = 6$  and calculate the two points of the ciphertext

$$C_1 = r \times e_1 = 6 \times (2, 1) = (8, 7)$$

$$C_2 = (4, 2) + 6 \times (4, 9) = (4, 2) + (8, 4) = (2, 10)$$

- f. Bob receives  $C_1$  and  $C_2$ . Bob calculates the plaintext as to get the  $P$ .

$$P = C_2 - (d \times C_1) = (2, 10) - 3 \times (8, 7)$$

$$= (2, 10) - (8, 4) = (2, 10) + (8, 7) = (4, 2)$$

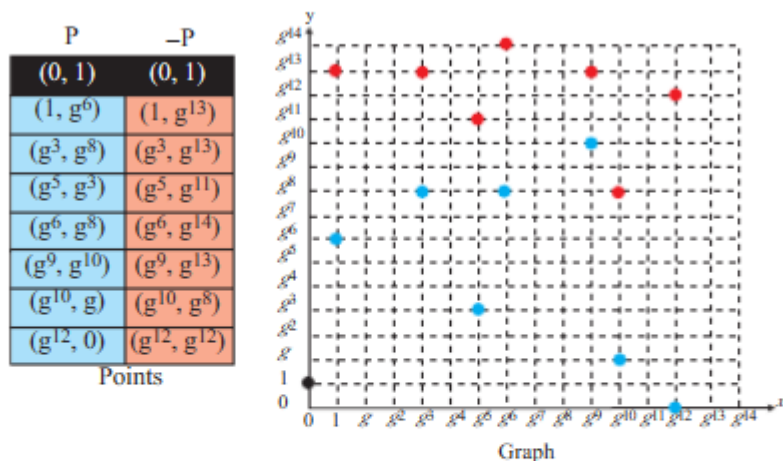
27.

- a.  $E(g^4, 1)$  means that  $a = g^4$  and  $b = 1$ . The equation of the curve is

$$y^2 + xy = x^3 + g^4x^2 + 1$$

- b. Assume that the irreducible polynomial is  $x^4 + x + 1$  (See Appendix G). We use the process in Example 10.16 in the textbook Page 326 to find the points on the curve as shown in Figure S10.27.

**Figure S10.27** Part of the solution to Exercise 27



- c. Bob chooses  $e_1 = (g^3, g^8)$  and  $d = 2$ . Then  $e_2 = d \times e_1 = (g^5, g^3)$ . The public key is the combination of  $e_1$ ,  $e_2$ , and  $E(g^4, 1)$ . The private key is  $d$ .
- d. Alice chooses  $P = (g^{10}, g)$  and  $r = 3$ .
- e. Alice calculates  $C_1 = r \times e_1 = (g^9, g^{10})$  and  $C_2 = P + r \times e_2 = (1, g^6)$ .
- f. Bob decrypt the message  $P = C_2 - (d \times C_1) = (g^{10}, g) = P$

28.

- a. The following shows the encrypting algorithm. It calls the **KnapsackSum** algorithm defined in the text (Algorithm 10.1).

```

KnapsackEncrypt ( $P, \mathcal{A}$ )
{
    C = KnapsackSum ( $P, \mathcal{A}$ )
    return C
}
    
```

- b. The following shows the decrypting algorithm. It calls the **inv\_KnapsackSum** algorithm defined in the text (Algorithm 10.1).

```

KnapsackDecrypt ( $C, \mathcal{A}, r, n$ )
{
     $C' = r^{-1} \times C \bmod n$ 
     $P' = \text{inv\_KnapsackSum}(C', \mathcal{A})$ 
     $P = \text{permute}(P')$ 
    return C
}
    
```

29.

- a. The following shows the encrypting algorithm. Alice uses the original message ( $m$ ), a random number ( $r$ ) and two functions (G and H).

```
RSA_OAEP_Encrypt ( $m, e, n, r$ )  
{  
     $M \leftarrow \text{pad}(m)$   
     $P_1 \leftarrow M \oplus G(r)$   
     $P_2 \leftarrow H(P_1) \oplus r$   
     $C \leftarrow \text{Fast\_Exponentiation}(P_1 \parallel P_2, e, n)$   
    return C  
}
```

- b. The following shows the decrypting algorithm. Bob uses the ciphertext (C), the private key ( $d$ ), modulus ( $n$ ), and two functions G and H.

```
RSA_OAEP_Decrypt ( $C, d, n$ )  
{  
     $P \leftarrow \text{Fast\_Exponentiation}(C, d, n)$   
     $P_1, P_2 \leftarrow \text{Split}_{m,k}(P)$   
     $r \leftarrow H(P_1) \oplus P_2$   
     $M \leftarrow P_1 \oplus G(r)$   
     $m \leftarrow \text{Extract}(M)$   
    return  $m$   
}
```

30.

```
CyclingAttack ( $C, e, n$ )  
{  
     $T_1 \leftarrow C$   
     $T_2 \leftarrow T_1^e \bmod n$   
    while ( $T_2 \neq C$ )  
    {  
         $T_1 \leftarrow T_2$   
         $T_2 \leftarrow T_1^e \bmod n$   
    }  
    return  $T_1$   
}
```