**UNIT-3(PART-I)**

**(NUMBER THEORY)**

**Prime Numbers**

An integer $p > 1$ is a prime number if and only if its only divisors[2] are $\pm 1$ and $\pm p$. **Prime numbers** play a critical role in number theory

Any integer $a > 1$ can be factored in a unique way as

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_t^{a_t} \quad (8.1)$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each $a_i$ is a positive integer. This is known as the fundamental theorem of arithmetic; a proof can be found in any text on number theory.

$$91 = 7 \times 13$$
$$3600 = 2^4 \times 3^2 \times 5^2$$
$$11011 = 7 \times 11^2 \times 13$$

It is useful for what follows to express this another way. If P is the set of all prime numbers, then any positive integer $a$ can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a_p} \qquad \text{where each } a_p \geq 0$$

**Relative Prime Numbers**

Two numbers are said to be relative prime numbers when they share no factors in common other than one

If two integers a, b are relatively prime if gcd(a,b)=1.

Examples:

1) 15 and 28 are relatively prime numbers
   15=the factors are (1,3,5)
   28=the factors are(1,2,4,7)
2) 7 and 20 are relatively prime numbers
3) 12 and 13 are relatively prime numbers

**MODULAR ARITHMETIC**

**The Modulus**

If 'a' is an integer and'n' is a positive integer, we define **amodn** to be the remainder when is 'a' is divided by 'n' . The integer is called the **modulus**. Thus, for any integer , we can write the Equation as follows:

$$\boldsymbol{a = qn + r}$$

example:11 mod 7 = 4; - 11 mod 7 = 3

**Congruence**

Two integers and are said to be **congruent modulo *n***, if (a mod n)= (*b* mod *n*).This is written as

$$a \equiv b \pmod{n}.$$

We say that a is congruent to b modulo m if m divides b-a or a-b.

Examples:

$73 \equiv 4 \pmod{23}; \qquad 21 \equiv -9 \pmod{10}$

$20 \equiv 0 \pmod{10}$

**Properties of Congruences**

Congruences have the following properties:

1. $a \equiv b \pmod{n}$ if $n|(a - b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$.
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

**Modular Arithmetic Operations**

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

Examples:

$11 \bmod 8 = 3; 15 \bmod 8 = 7$
$[(11 \bmod 8) + (15 \bmod 8)] \bmod 8 = 10 \bmod 8 = 2$
$(11 + 15) \bmod 8 = 26 \bmod 8 = 2$

$[(11 \bmod 8) - (15 \bmod 8)] \bmod 8 = -4 \bmod 8 = 4$
$(11 - 15) \bmod 8 = -4 \bmod 8 = 4$

$[(11 \bmod 8) \times (15 \bmod 8)] \bmod 8 = 21 \bmod 8 = 5$
$(11 \times 15) \bmod 8 = 165 \bmod 8 = 5$

**Modular Addition And Multiplication**

Table 4.2 provides an illustration of modular addition and multiplication modulo 8

Table 4.2 Arithmetic Modulo 8

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 |
| 3 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 |
| 4 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| 5 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 |
| 6 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 |
| 7 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

(a) Addition modulo 8

| × | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 0 | 2 | 4 | 6 | 0 | 2 | 4 | 6 |
| 3 | 0 | 3 | 6 | 1 | 4 | 7 | 2 | 5 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 0 | 4 |
| 5 | 0 | 5 | 2 | 7 | 4 | 1 | 6 | 3 |
| 6 | 0 | 6 | 4 | 2 | 0 | 6 | 4 | 2 |
| 7 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

(b) Multiplication modulo 8

**Properties of Modular Arithmetic**

Define the set $Z_n$ as the set of nonnegative integers less than $n$:

$$Z_n = \{0, 1, \ldots, (n-1)\}$$

This is referred to as the **set of residues**, or **residue classes** (mod $n$). To be more precise, each integer in $Z_n$ represents a residue class. We can label the residue classes (mod $n$) as $[0], [1], [2], \ldots, [n-1]$, where

$$[r] = \{a: a \text{ is an integer}, a \equiv r \pmod{n}\}$$

The residue classes (mod 4) are

$[0] = \{\ldots, -16, -12, -8, -4, 0, 4, 8, 12, 16, \ldots\}$
$[1] = \{\ldots, -15, -11, -7, -3, 1, 5, 9, 13, 17, \ldots\}$
$[2] = \{\ldots, -14, -10, -6, -2, 2, 6, 10, 14, 18, \ldots\}$
$[3] = \{\ldots, -13, -9, -5, -1, 3, 7, 11, 15, 19, \ldots\}$

Fermat's Theorem:-

This theorem state the following. If 'p' is prime and 'a' is positive integer and not divisible by 'p' then $a^{p-1} \equiv 1 \pmod{p}$ and every integer 'a' $a^p \equiv a \pmod{p}$

This means that if you divide $a^p$ by p then the remainder is 'a'.

It is useful in public key cryptography (RSA) and pramality testing.

Proof:- Consider a set of positive integers lessthan 'p'

$$A = \{1, 2 \ldots (p-1)\}$$

Multiply each element 'a' and "modulo p" then we can get

$$X = \{a \bmod p, 2a \bmod p, \ldots (p-1)a \bmod p\}$$

None of the element of 'x' is equal to 0 because 'p' doesnot divide 'a'

No two of the integers in 'x' are equal.

Assume that $ja \equiv ka \pmod{p}$

where $1 \leq j < k \leq p-1$

Eliminate 'a' from both sides because 'a' is relative prime to 'p'

$$j \equiv k \pmod{p}$$

This is impossible. J and k both are positive integers lessthan 'p'. we know that p-1 elements of x are all positive integers. we can conclude the 'x' consists of a set of integers $\{1, 2 \ldots p-1\}$

Multiply the numbers in both sets and taking the result mod p produces

$$a * 2a \ldots (p-1)a = [(1 * 2 \ldots (p-1)] \pmod{p}$$

$$a^{p-1}(p-1)! = (p-1)! \pmod p$$

cancel "(p-1)!" because it is relative prime to p

$$\boxed{a^{p-1} \equiv 1 \pmod p} \rightarrow ①$$

An alternative form of formals theorem is given as

$$\boxed{a^p = a \pmod p} \rightarrow ②$$

Eg:- $(6^{17}) \bmod 13$

$(6^{12} \times 6^5) \bmod 13$

$(6^{12} \bmod 13) * (6^5 \bmod 13)$

$= 1 * 6^5 \bmod 13$

$= 2$

Eg:- $2^{48} \bmod 5$

$= 1$

a) Find the least residue of $9^{794}$ modulo 73

73 is prime

from fermats theorem $9^{72} \equiv 1 \bmod 73$

$9^{794} = (9^{72})^{11} \cdot 9^2 \equiv 9^2 \pmod{73}$

$\equiv 81 \pmod{73}$

$= 8$

b) Find the solution to the congruence $x^{86} \equiv 6 \pmod{29}$

29 is prime

$x^{28} \equiv 1 \pmod{29} \rightarrow ①$

$x^{86} = (x^{28})^3 \cdot x^2$

$x^2 \equiv 6 \pmod{29}$

x=8

(d) Use fermat theorem to find the number 'x' between 0 and to 28 · $x^{85}$ congruence to 6 (mod 35)

$x^{85} \equiv 6 \pmod{35}$

$x^{34} \equiv 1 \pmod{35}$

No solution

35 is not prime
So according to "fermat's" theorem this can ~~not~~ be not solved in

Euler's Theorem:

Euler's totient function:-

Before going to euler's theorem we must know about important quantity in number theorey ie Euler's totient function. It can be represented as $\phi$

$\phi(n)$ is the number of positive integers lessthan 'n' and the relative prime to 'n'.

Means how many numbers that are between 1 and (n-1) that are relative prime to n.

$\phi(35)$ : 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

$\phi(35) = 24$

$\phi(37) = 36$

$\phi(p) = (p-1)$

It should clear that for a prime number p. $\phi(p) = (p-1)$

By this $\phi(n)$ will be easy to calculate when 'n' has exactly two different prime factors.

That is we have two prime numbers p and q with $p \neq q$ the $n = pq$.

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q)$$
$$= (p-1) \times (q-1)$$

Ex: $\phi(21) = \phi(3 \times 7) = \phi(3) \times \phi(7)$

$= 2 \times 6 = 12$

## Euler's Theorem:

Euler's theorem states that for every 'a' and 'n' the are relatively prime

$$a^{\phi(n)} = 1 \pmod{n} \longrightarrow (1)$$

1. $a=3, n=10$, show that $a^{\phi(n)} \equiv 1 \pmod{n}$

$$3^{\phi(10)} = 1 \pmod{10} \quad [\phi(10) = \phi(5\times 2) = 4\times 1 = 4]$$

$$3^4 = 1 \pmod{10}$$

$$81 = 1 \pmod{10}$$

2. $a=2, n=11$ then show that $a^{\phi(n)} \equiv 1 \pmod{n}$

$$2^{\phi(11)} \equiv 1 \pmod{n}$$

$$2^{10} = 1 \pmod{11}$$

Proof:- Above equation (1) is true if 'n' is prime because in that case $\phi(n) = n-1$ and fermat's theorem holds.

From Euler's totient function $\phi(n)$ is the number of positive integers less than 'n' that are relative prime to 'n'

Consider a set of such integers labeled as $R = \{x_1, x_2, \ldots x_{\phi(n)}\}$

Now multiply each element by 'a', modulo 'n'

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n) \cdots (ax_{\phi(n)} \bmod n)\}$$

The set 's' is a permutation of 'r' by the following reasons.

(1) 'a' is a relative prime to 'n' and $x_i$ is a relative prime to 'n', $ax_i$ must also be relative prime to 'n'. Thus all the numbers of s are integers less than 'n' that are relative prime to 'n'.

(2) There are no duplicates in 's'

if $a x_i \bmod n = a x_j \bmod n$

$$\therefore \frac{\phi(n)}{\pi} (a x_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} a x_i = \prod_{i=1}^{\phi(n)} x_i \pmod n$$

→ eliminate

$$a^{\phi(n)} \times \left[ \prod_{i=1}^{\phi(n)} \right] x_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)} = 1 \bmod n$$

CRT (chinese Remainder Theorem):-

→ One of the most useful elements of number theory is CRT.

→ CRT says it is possible to reconstructs integers in a certain range in their residues module, a set of pairwise relatively prime module

→ The CRT is a method for solving certain system of congruency

→ The CRT is a mechanism for manipulating very large numbers in terms of tuples of small integers.

Theorem:- Suppose that $m_1, m_2 \ldots m_r$ are relative prime numbers and let $a_1, a_2 \ldots a_r$ be integers then the system of congruences

$x \equiv a_i \pmod{m_i}$ for $1 \le i \le r$ has a unique solution modulo $M = m_1 \times m_2 \times \ldots \times m_r$ which is given by

$x \equiv a_1 m_1 y_1 + a_2 m_2 y_2 + \ldots + a_r m_r y_r \pmod m$ where

Explanation $M_i = M/m_i$ and $Y_i \equiv (M_i)^{-1} \pmod{m_i}$ for $1 \le i \le r$

The CRT can be evaluated in five steps.

$$X \equiv a_1 \pmod{m_1}$$
$$X \equiv a_2 \pmod{m_2}$$
$$X \equiv a_3 \pmod{m_3}$$

$$X \equiv a_i \pmod{m_i}$$

Step1:- calculate $M = m_1 \times m_2 \cdots \times m_i$ ie. $\prod_{K=1}^{i} m_k$

Step2:- calculate $M_1 = M/m_1$, $M_2 = M/m_2 \cdots M_i = M/m_i$

Step3:- calculate A's = $\{a_1, a_2, a_3 \cdots a_i\}$ if necessary

Step4:- Find multiplicative inverse of each $M_1, M_2, \ldots m_i$ mod $m_1, m_2 \cdots m_i$ respectively

Step5:- $\sum (a_i M_i M_i^{-1}) \bmod M$ is result

Examples: Refer the examples given in the class room.

**DISCRETE LOGARITHMS**

Multiplicative order:-

Given an integer 'a' and positive integer 'n' with gcd (a,n)=1 (a ,n are relative prime)

The multiplicative order of a 'modulo n' is smallest positive integer 'k' with $a^k \equiv 1 \pmod n$

The order of modulo 'n' is written as $ord_n(a)$ or $O_n(a)$

Ex: Define multiplicative order of 4 mod 7.

$4^2 = 16 \equiv 2 \pmod 7$

$4^3 = 64 \equiv 1 \pmod 7$

$ord_7(4) = 3$

Find the multiplicative order of 2 mod 7

$2 \equiv 2 \pmod 7$

$2^2 \equiv 4 \pmod 7$

$2^3 \equiv 1 \pmod 7$

$Ord_7(2) = 3$

**Primitive root**

If a is a primitive root of n then its powers

$$a, a^2, \ldots, a^{\phi(n)}$$

are distinct (mod $n$) and are all relatively prime to $n$. In particular, for a prime number $p$, if $a$ is a primitive root of $p$, then

$$a, a^2, \ldots, a^{p-1}$$

are distinct (mod $p$). For the prime number 19, its primitive roots are 2, 3, 10, 13, 14, and 15.

Exaple: : Refer the examples given in the class room

**Logarithms in modular arithmetic:**

by the definition of modular arithmetic. It follows that for any integer $b$ and a primitive root $a$ of prime number $p$, we can find a unique exponent $i$ such that

$$b \equiv a^i (\text{mod } p) \quad \text{where } 0 \leq i \leq (p-1)$$

This exponent $i$ is referred to as the **discrete logarithm** of the number $b$ for the base $a$ (mod $p$). We denote this value as $\text{dlog}_{a,p}(b)$.[10]

Examples: refer examples given in the class room

Note:solve the problems on Fermat's Thoerm and Eulers Theorm(given in the class room)

**UNIT-3(PART-II)**

**PUBLIC KEY CRYPTOGRAPHY**

## PUBLIC KEY CRYPTOSYSTEM PRINCIPLES:

- Public key cryptography also Known as **asymmetric cryptography**.
- Public-key systems depend on the use of mathematical functions.
- The concept of public key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption.
    - The Key Exchange Problem
    - The Trust Problem

    **The Key Exchange Problem:** The key exchange problem arises from the fact that communicating parties must somehow share a secret key before any secure communication can be initiated, and both parties must then ensure that the key remains secret. Of course, direct key exchange is not always feasible due to risk, inconvenience, and cost factors.

    **The Trust Problem:** Ensuring the integrity of received data and verifying the identity of the source of that data can be very important. Means in the symmetric key cryptography system, receiver doesn't know whether the message is coming for particular sender.
- This public key cryptosystem uses two keys as pair for encryption of plain text and Decryption of cipher text.
- These two keys are names as "**Public key**" and "**Private key**". The private key is kept secret whereas public key is distributed widely.
- A message or text data which is encrypted with the public key can be decrypted only with the corresponding private-key
- This two key system very useful in the areas of confidentiality (secure) and authentication

| A **public-key encryption** scheme has **six ingredients** | | |
|---|---|---|
| 1 | **Plaintext** | This is the readable message or data that is fed into the algorithm as input. |
| 2 | **Encryption algorithm** | The encryption algorithm performs various transformations on the plaintext. |

| | | |
|---|---|---|
| 3 | **Public key** | This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input |
| 4 | **Private key** | |
| 5 | **Ciphertext** | This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts. |
| 6 | **Decryption algorithm** | This algorithm accepts the ciphertext and the matching key and produces the original plaintext. |

**Public key cryptography for providing confidentiality (secrecy)**

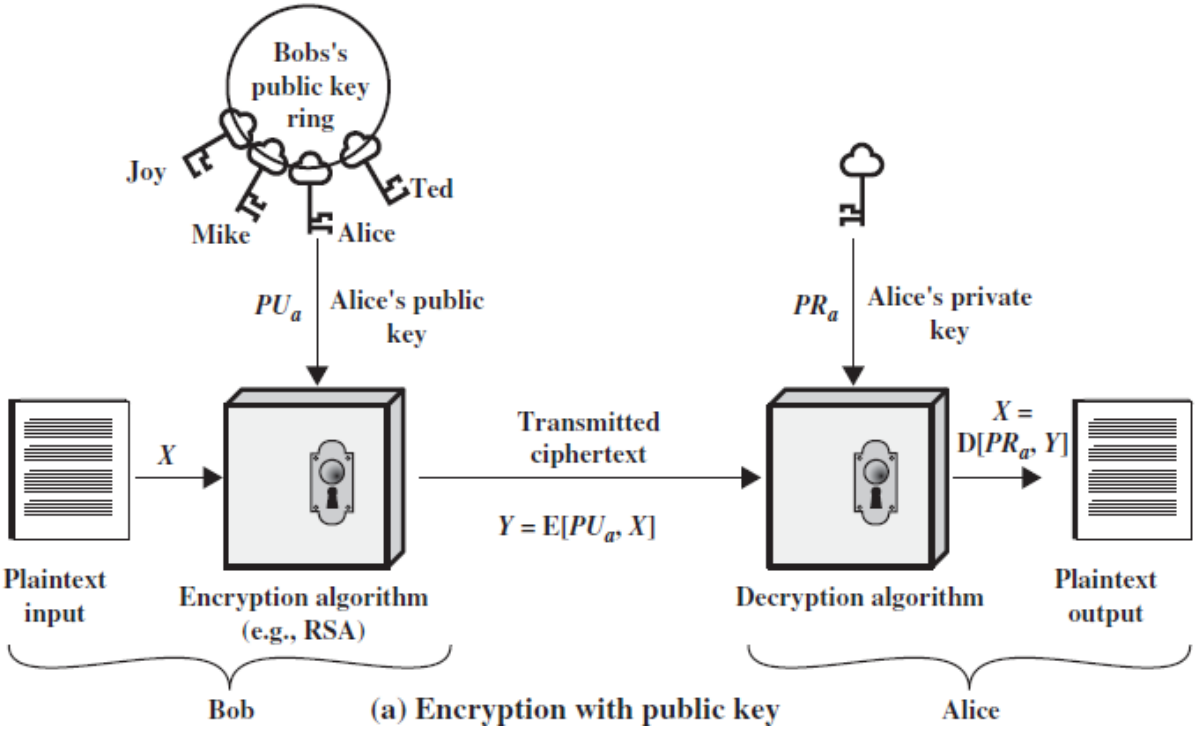



**Figure 9.1**

**The essential steps are the following.**

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key.The companion key is kept private. As Figure 9.1a suggests, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

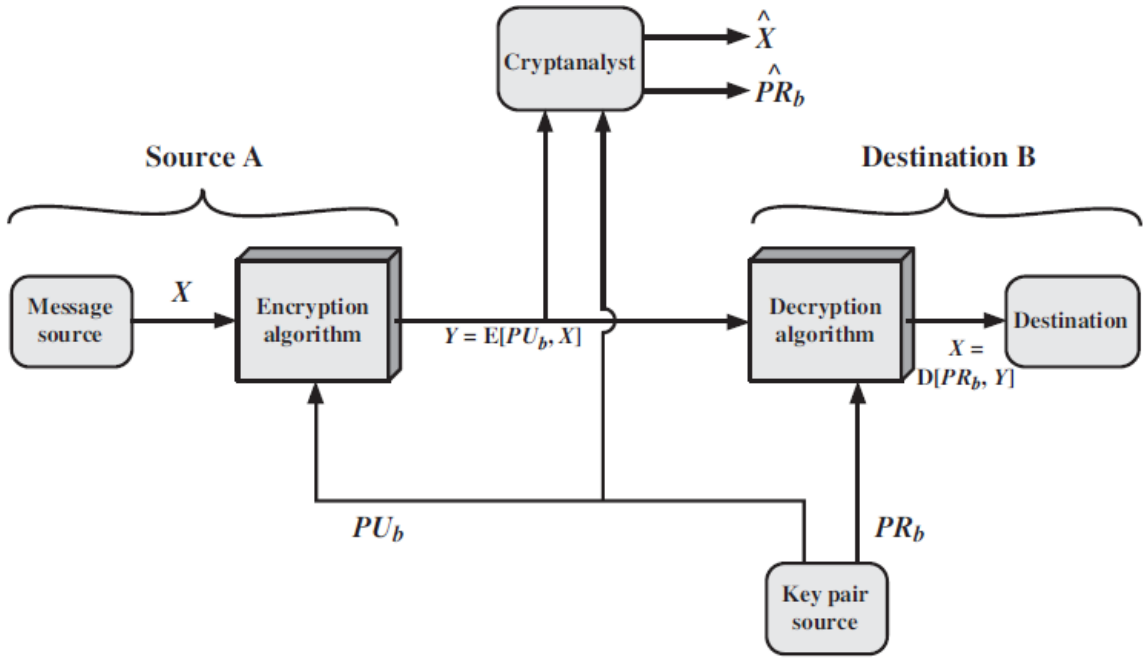



Figure 9.2 Public-Key Cryptosystem: Secrecy

There is some source A that produces a message in plaintext ***X* = [*X*1, *X*2, . . . ,X*M*].**

The ***M*** elements of ***X*** are letters in some finite alphabet. The message is intended for destination **B**.

B generates a related pair of keys: a public key, $PU_b$, and a private key, $PR_b$.

$PR_b$ is known only to B, whereas $PU_b$ is publicly available and therefore accessible by A.

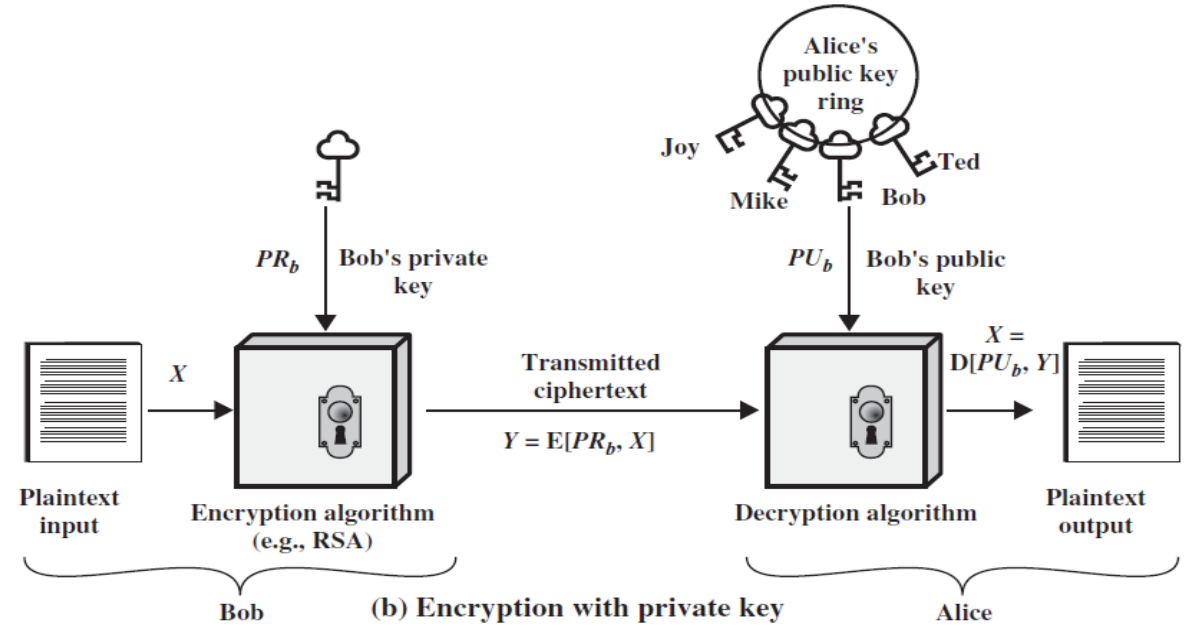
With the message $X$ and the encryption key $PU_b$ as input, A forms the ciphertext $Y = [Y1, Y2, \ldots, Y_N]$:

$$Y = \mathrm{E}(PU_b, X)$$

The intended receiver, in possession of the matching private key, is able to invert the transformation:

$$X = \mathrm{D}(PR_b, Y)$$

**Public key cryptography for proving Authentication:**




(b) Encryption with private key

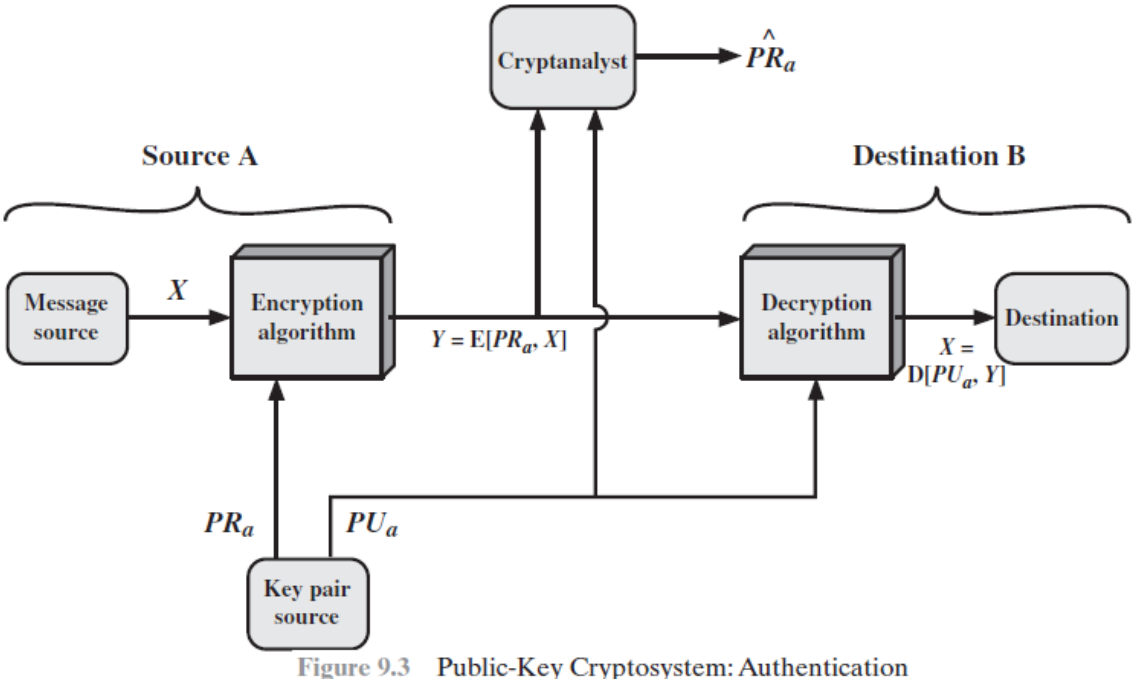



Figure 9.3 Public-Key Cryptosystem: Authentication

The above diagrams show the use of public-key encryption to provide authentication:

$$Y = E(PR_a, X)$$

$$X = D(PU_a, Y)$$

- In this case,A prepares a message to B and encrypts it using A's private key before transmitting it. B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message. Therefore, the entire encrypted message serves as a **digital signature.**

- It is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.

**Applications for Public-Key Cryptosystems**

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function.

**We can classify the use of public-key cryptosystems into three categories**

**• Encryption /decryption:** The sender encrypts a message with the recipient's public key.

**• Digital signature:** The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

**• Key exchange:** Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

Applications for Public-Key Cryptosystems

| **Algorithm** | **Encryption/Decryption** | **Digital Signature** | **Key Exchange** |
|---|---|---|---|
| RSA | Yes | Yes | Yes |
| Elliptic Curve | Yes | Yes | Yes |
| Diffie-Hellman | No | No | Yes |
| DSS | No | Yes | No |

**Public-Key Cryptanalysis**

- As with symmetric encryption, a public-key encryption scheme is vulnerable to a brute-force attack. The countermeasure is the same: Use large keys
- Public-key systems depend on the use of some sort of invertible mathematical function.

- Thus, the key size must be large enough to make brute-force attack impractical but small enough for practical encryption and decryption. In practice, the key sizes that have been proposed do make brute-force attack impractical but result in encryption/decryption speeds that are too slow for general-purpose use.
- Instead, as was mentioned earlier, public-key encryption is currently confined to key management and signature applications.

## RSA ALGORITHM.

- It is one of the most common public key algorithm.
- It was first published by Rivest (R), Shamir (S) and Adleman (A) in the year 1977.
- The RSA scheme is a block cipher in which the plaintext & ciphertext are integers between 0 and n-1 for some 'n'.
- A typical size for 'n' is 1024 bits or 309 decimal digits. That is, n is less than $2^{1024}$

**Description of the Algorithm:**

- RSA algorithm uses an expression with exponentials.
- In RSA plaintext is encrypted in blocks, with each block having a binary value less than some number n. that is, the block size must be less than or equal to **$\log_2(n)$**
- **RSA** uses two exponents 'e' and'd' where e→public and d→private.
- Encryption and decryption are of following form, for some PlainText 'M' and Ciphertext block 'C'

$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

$$M = C^d \bmod = (M^e \bmod n)^d \text{ mon } n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the value of n.

The sender knows the value of 'e' & only the reviver knows the value of 'd' thus this is a public key encryption algorithm with a

Public key PU={e, n}

Private key PR={d, n}

**Requirements:**

The RSA algorithm to be satisfactory for public key encryption, the following requirements must be met:

1. It is possible to find values of e, d n such that " $\mathbf{M^{ed} \bmod n = M}$ " for all M<n
2. It is relatively easy to calculate " $\mathbf{M^e \bmod n}$ " and " $\mathbf{C^d \bmod n}$ "for M<n
3. It is infeasible to determine "d" given 'e' & 'n'. The " $\mathbf{M^{ed} \bmod n = M}$ "

Then the relation between 'e' & 'd' can be expressed as

$$\boxed{ed \bmod \emptyset(n)=1}$$

this is equivalent to saying

$$\boxed{ed \equiv 1 \bmod \emptyset(n)}$$
$$\boxed{d \equiv e^{-1} \bmod \emptyset(n)}$$

That is 'e' and 'd' are multiplicative inverses mod Ø(n).

**Steps of RSA algorithm:**

Step 1→Select 2 prime numbers p & q

Step 2→Calculate n=pq

Step 3→Calculate Ø(n)=(p-1)(q-1)

Step 4→ Select or find integer e (public key) which is relatively prime to Ø(n).

ie., gcd (Ø(n), e)=1 where 1<e< Ø(n).

Step 5→ Calculate "d" (private key) by using following condition. $\boxed{ed \equiv 1 \bmod \emptyset(n)}$

d< Ø(n).

Step 6→ Perform encryption by using $\boxed{C = M^e \bmod n}$

Step 7→ perform Decryption by using $\boxed{M = C^d \bmod n}$

Figure 9.5 summarizes the RSA algorithm. If Bob wants to send a message to Alice,Alice generates a public/private key pair; Bob encrypts using Alice's public key; and Alice decrypts using her private key.

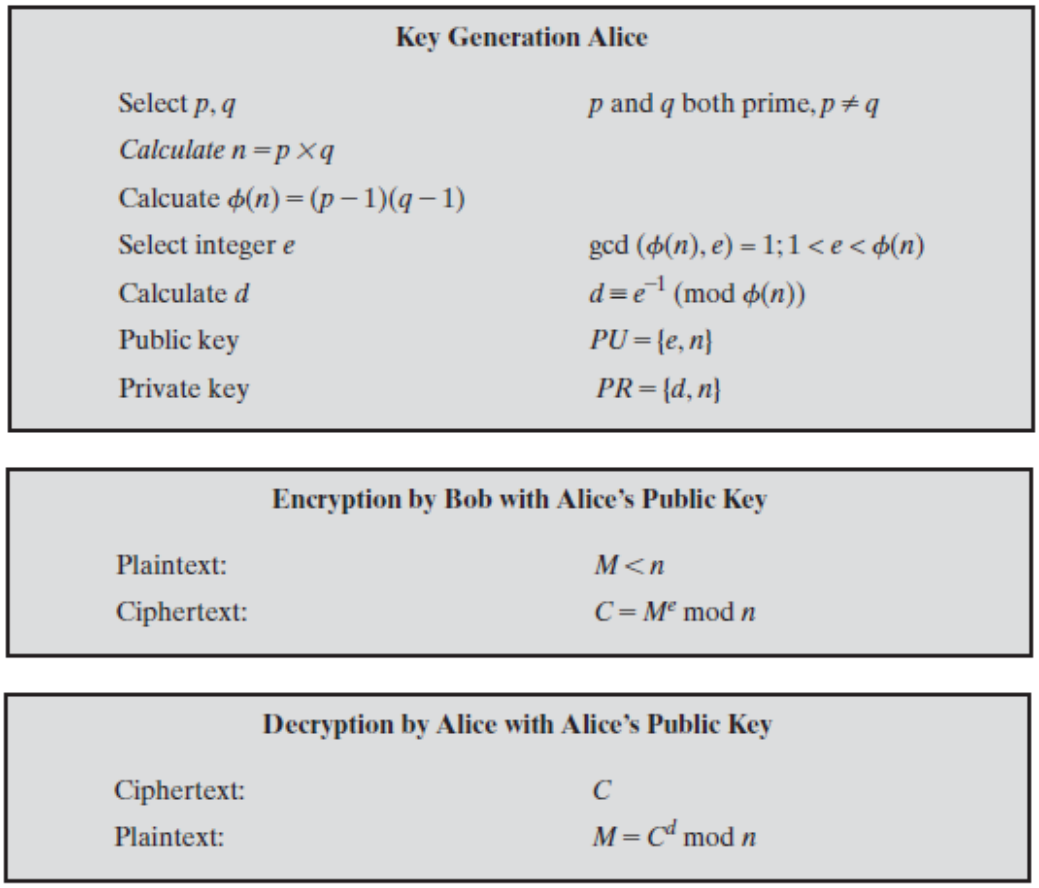



Figure 9.5 The RSA Algorithm

**Example:**

**1.** Select two prime numbers, $p = 17$ and $q = 11$.

**2.** Calculate $n = pq = 17 \times 11 = 187$.

**3.** Calculate $\emptyset(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.

**4.** Select $e$ such that $e$ is relatively prime to $\emptyset(n) = 160$ and less than $\emptyset(n)$; we choose $e = 7$.

**5.** Determine $d$ such that $de \equiv 1 \pmod{160}$ and $d < 160$.The correct value is $d = 23$, because $23 * 7 = 161 = (1 \times 160) + 1$; $d$ can be calculated using the extended Euclid's algorithm

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$.

The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows.

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$$

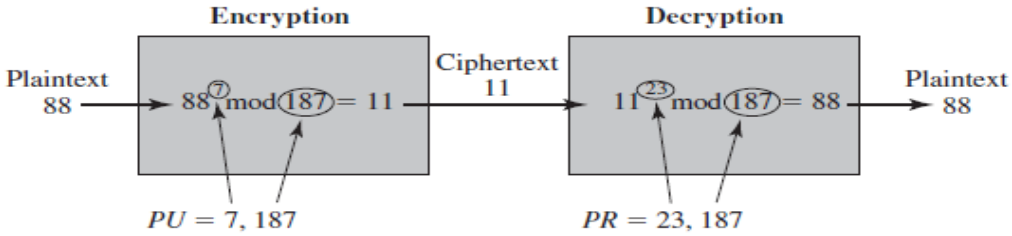



Figure 9.6 Example of RSA Algorithm

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$$

$$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$$

$$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$$

**Example 2:**

- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute φ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
- Choose e such that 1 < e < φ(n) and e and n are coprime. Let e = 7
- Compute a value for d. One solution is d = 3 [(3 * 7) % 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- Consider the plain text **m=2**
- The encryption of m = 2 is c = $2^7$ mod33= 29
- The decryption of c = 29 is m = $29^3$mod33 **= 2**

**The Security of RSA**

There are three main approaches of attacking RSA algorithm.

• **Brute force:** This involves trying all possible private keys.

• **Mathematical attacks:** There are several approaches, all equivalent in effort to factoring the product of two primes.

• **Timing attacks:** These depend on the running time of the decryption algorithm.

• **Chosen ciphertext attacks:** This type of attack exploits properties of the RSA algorithm.

**DIFFIE-HELLMAN KEY EXCHANGE:**

- Diffie-Hellman key exchange is the first published public key algorithm
- This Diffie-Hellman key exchange protocol is also known as exponential key agreement. And it is based on mathematical principles.
- The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages.
- This algorithm itself is limited to exchange of the keys.
- This algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.
- The discrete logarithms are defined in this algorithm in the way of define a primitive root of a prime number.
    - Primitive root: we define a primitive root of a prime number P as one whose power generate all the integers form 1 to P-1 that is if 'a' is a primitive root of the prime number P, then the numbers

      $a \bmod P, a^2 \bmod P, a^3 \bmod P, \ldots\ldots a^{P-1} \bmod P$ are distinct and consist of the integers form 1 through P-1 in some permutation.

      For any integer 'b' and 'a', here 'a' is a primitive root of prime number P, then

      $\mathbf{b \equiv a^i \bmod P \quad 0 \le i \le (P-1)}$

The exponent i → is refer as discrete logarithm or index of b for the base a, mod P.

The value denoted as $\mathbf{ind_{a,p}(b)}$

**Algorithm for Diffie-Hellman Key Exchange:**

Step 1→ consider two publicly known numbers q, α

q→ Prime number

α→ primitive root of q and α< q.

Step 2 → if A & B users wish to exchange a key

a) User A select a random integer $X_A$<q and computes

$$Y_A = \alpha^{X_A} \bmod q$$

b) User B independently select a random integer $X_B$ <q and computes

$$Y_B = \alpha^{X_B} \bmod q$$

c) Each side keeps the X value private and Makes the Y value available publicly to the outer side.

Step 3→ User A Computes the key as

$$K = (Y_B)^{X_A} \bmod q$$

User B Computes the key as

$$K = (Y_A)^{X_B} \bmod q$$

Step 4→ two calculation produce identical results

$$K = (Y_B)^{X_A} \bmod q$$

$$K = (\alpha^{X_B} \bmod q)^{X_A} \bmod q \quad \text{(We know that } Y_B = \alpha^{X_B} \bmod q \text{)}$$

$$= (\alpha^{X_B})^{X_A} \bmod q$$

$$= (\alpha^{X_A})^{X_B} \bmod q$$

$$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$$

$$= (Y_A)^{X_B} \bmod q \quad \text{(We know that } Y_A = \alpha^{X_A} \bmod q \text{)}$$

The result is that the two sides have exchanged a secret key.

| Global Public Elements | |
|---|---|
| $q$ | prime number |
| $\alpha$ | $\alpha < q$ and $\alpha$ a primitive root of $q$ |

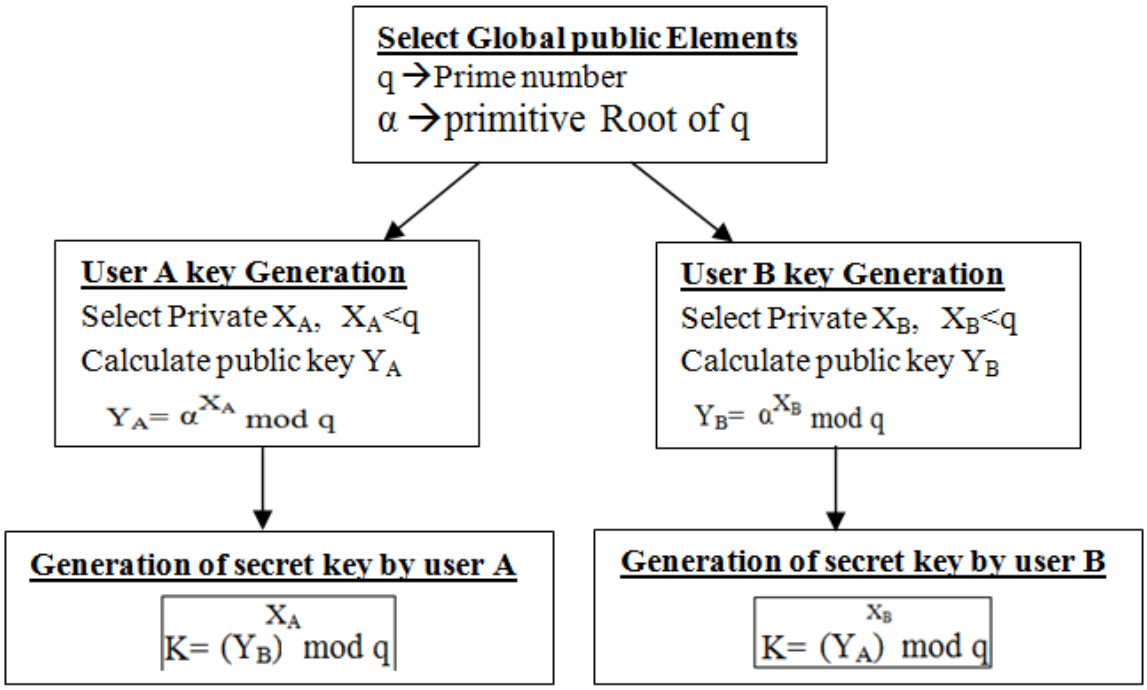
| User A Key Generation | |
|---|---|
| Select private $X_A$ | $X_A < q$ |
| Calculate public $Y_A$ | $Y_A = \alpha^{XA} \bmod q$ |

| User B Key Generation | |
|---|---|
| Select private $X_B$ | $X_B < q$ |
| Calculate public $Y_B$ | $Y_B = \alpha^{XB} \bmod q$ |

| Calculation of Secret Key by User A |
|---|
| $K = (Y_B)^{XA} \bmod q$ |

| Calculation of Secret Key by User B |
|---|
| $K = (Y_A)^{XB} \bmod q$ |

Figure 10.1 The Diffie-Hellman Key Exchange Algorithm




**Example:**

Here is an example. Key exchange is based on the use of the prime number $q = 353$ and a primitive root of 353, in this case $\alpha = 3$. A and B select secret keys $X_A = 97$ and $X_B = 233$, respectively. Each computes its public key:

A computes $Y_A = 3^{97} \bmod 353 = 40$.

B computes $Y_B = 3^{233} \bmod 353 = 248$.

After they exchange public keys, each can compute the common secret key:

A computes $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$.

B computes $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$.

## ELGAMAL CRYPTOGRAPHIC SYSTEM

- Elgamal is a a public-key scheme based on discrete logarithms,
- It is closely related to the Diffie-Hellman technique
- It is used in digital signature standard (DSS), and the S/MIME e-mail standard

As with Diffie-Hellman, the global elements of ElGamal are a prime number $q$ and $\alpha$, which is a primitive root of $q$. User A generates a private/public key pair as follows:

1. Generate a random integer $X_A$, such that $1 < X_A < q - 1$.
2. Compute $Y^A = \alpha^{X_A} \bmod q$.
3. A's private key is $X_A$; A's pubic key is $\{q, \alpha, Y_A\}$.

Any user B that has access to A's public key can encrypt a message as follows:

1. Represent the message as an integer $M$ in the range $0 \leq M \leq q - 1$. Longer messages are sent as a sequence of blocks, with each block being an integer less than $q$.
2. Choose a random integer $k$ such that $1 \leq k \leq q - 1$.
3. Compute a one-time key $K = (Y_A)^k \bmod q$.
4. Encrypt $M$ as the pair of integers $(C_1, C_2)$ where

$$C_1 = \alpha^k \bmod q; \quad C_2 = KM \bmod q$$

User A recovers the plaintext as follows:

1. Recover the key by computing $K = (C_1)^{X_A} \bmod q$.
2. Compute $M = (C_2 K^{-1}) \bmod q$.

These steps are summarized in Figure 10.3. It corresponds to Figure 9.1a: Alice generates a public/private key pair; Bob encrypts using Alice's public key; and Alice decrypts using her private key.

Let us demonstrate why the ElGamal scheme works. First, we show how $K$ is recovered by the decryption process:

| | |
|---|---|
| $K = (Y_A)^k \bmod q$ | $K$ is defined during the encryption process |
| $K = (\alpha^{X_A} \bmod q)^k \bmod q$ | substitute using $Y_A = \alpha^{X_A} \bmod q$ |
| $K = \alpha^{kX_A} \bmod q$ | by the rules of modular arithmetic |
| $K = (C_1)^{X_A} \bmod q$ | substitute using $C_1 = \alpha^k \bmod q$ |

Next, using $K$, we recover the plaintext as

$$C_2 = KM \bmod q$$

$$(C_2 K^{-1}) \bmod q = KMK^{-1} \bmod q = M \bmod q = M$$

We can restate the ElGamal process as follows, using Figure 10.3.

1. Bob generates a random integer $k$.
2. Bob generates a one-time key $K$ using Alice's public-key components $Y_A$, $q$, and $k$.
3. Bob encrypts $k$ using the public-key component $\alpha$, yielding $C_1$. $C_1$ provides sufficient information for Alice to recover $K$.
4. Bob encrypts the plaintext message $M$ using $K$.
5. Alice recovers $K$ from $C_1$ using her private key.
6. Alice uses $K^{-1}$ to recover the plaintext message from $C_2$.

| Global Public Elements | |
|---|---|
| $q$ | prime number |
| $\alpha$ | $\alpha < q$ and $\alpha$ a primitive root of $q$ |

| Key Generation by Alice | |
|---|---|
| Select private $X_A$ | $X_A < q - 1$ |
| Calculate $Y_A$ | $Y_A = \alpha^{XA} \bmod q$ |
| Public key | $PU = \{q, \alpha, Y_A\}$ |
| Private key | $X_A$ |

| Encryption by Bob with Alice's Public Key | |
|---|---|
| Plaintext: | $M < q$ |
| Select random integer $k$ | $k < q$ |
| Calculate $K$ | $K = (Y_A)^k \bmod q$ |
| Calculate $C_1$ | $C_1 = \alpha^k \bmod q$ |
| Calculate $C_2$ | $C_2 = KM \bmod q$ |
| Ciphertext: | $(C_1, C_2)$ |

| Decryption by Alice with Alice's Private Key | |
|---|---|
| Ciphertext: | $(C_1, C_2)$ |
| Calculate $K$ | $K = (C_1)^{XA} \bmod q$ |
| Plaintext: | $M = (C_2 K^{-1}) \bmod q$ |

Figure 10.3 The ElGamal Cryptosystem

---

For example, let us start with the prime field GF(19); that is, $q = 19$. It has primitive roots {2, 3, 10, 13, 14, 15}, as shown in Table 8.3. We choose $\alpha = 10$.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 5$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$ (see Table 8.3).
3. Alice's private key is 5; Alice's pubic key is $\{q, \alpha, Y_A\} = \{19, 10, 3\}$.

---

Suppose Bob wants to send the message with the value $M = 17$. Then,

1. Bob chooses $k = 6$.
2. Then $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$.
3. So
   $C_1 = \alpha^k \bmod q = \alpha^6 \bmod 19 = 11$
   $C_2 = KM \bmod q = 7 \times 17 \bmod 19 = 119 \bmod 19 = 5$
4. Bob sends the ciphertext $(11, 5)$.

For decryption:

1. Alice calculates $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$.
2. Then $K^{-1}$ in GF(19) is $7^{-1} \bmod 19 = 11$.
3. Finally, $M = (C_2 K^{-1}) \bmod q = 5 \times 11 \bmod 19 = 55 \bmod 19 = 17$.

---

## ELLIPTIC CURVE CRYPTOGRAPHY

- **Definition: Elliptic curve cryptography (ECC)** is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. These are analogy of existing public key cryptosystem in which modular arithmetic is replaced by operations defined over elliptic curve.

## ECC DIFFIE-HELLMAN KEY EXCHANGE:

ECC can do key exchange, that is analogous to Diffie Hellman.

Key exchange using elliptic curves can be done in the following manner.

First pick a large integer q , which is either a prime number P or an integer of the form $2^m$ and elliptic curve parameters a & b for equation

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \quad \text{or} \quad y^2 + xy = x^3 + ax^2 + b .$$

This define elliptic group of point $E_q(a,b)$.

Pick a base point G=(x1,y1) in $E_p(a,b)$ whose order is a very large value n.

The order n of a point G on an elliptic curve is the smallest +ve integer n such that $nG=0.E_q(a,b)$

**Global Public Elements**

| | |
|---|---|
| $E_q(a, b)$ | elliptic curve with parameters $a$, $b$, and $q$, where $q$ is a prime or an integer of the form $2^m$ |
| $G$ | point on elliptic curve whose order is large value $n$ |

**User A Key Generation**

| | |
|---|---|
| Select private $n_A$ | $n_A < n$ |
| Calculate public $P_A$ | $P_A = n_A \times G$ |

**User B Key Generation**

| | |
|---|---|
| Select private $n_B$ | $n_B < n$ |
| Calculate public $P_B$ | $P_B = n_B \times G$ |

**Calculation of Secret Key by User A**

$K = n_A \times P_B$

**Calculation of Secret Key by User B**

$K = n_B \times P_A$

Figure 10.7 ECC Diffie-Hellman Key Exchange

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$$

As an example,[6] take $p = 211$; $E_p(0, -4)$, which is equivalent to the curve $y^2 = x^3 - 4$; and $G = (2, 2)$. One can calculate that $240G = O$. A's private key is $n_A = 121$, so A's public key is $P_A = 121(2, 2) = (115, 48)$. B's private key is $nB = 203$, so B's public key is $203(2, 3) = (130, 203)$. The shared secret key is $121(130, 203) = 203(115, 48) = (161, 69)$.

**Elliptic Curve Encryption/Decryption:**

Several approaches to encryption/decryption using elliptic curves have been analyzed in the literature. In this subsection, we look at perhaps the simplest. The first task in this system is to encode the plaintext message $m$ to be sent as an $x$–$y$ point $P_m$. It is the point $P_m$ that will be encrypted as a ciphertext and subsequently decrypted. Note that we cannot simply encode the message as the $x$ or $y$ coordinate of a point, because not all such coordinates are in $E_q(a, b)$; for example, see Table 10.1. Again, there are several approaches to this encoding, which we will not address here, but suffice it to say that there are relatively straightforward techniques that can be used.

As with the key exchange system, an encryption/decryption system requires a point $G$ and an elliptic group $E_q(a, b)$ as parameters. Each user A selects a private key $n_A$ and generates a public key $P_A = n_A \times G$.

To encrypt and send a message $P_m$ to B, A chooses a random positive integer $k$ and produces the ciphertext $C_m$ consisting of the pair of points:

$$C_m = \{kG, P_m + kP_B\}$$

Note that A has used B's public key $P_B$. To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

A has masked the message $P_m$ by adding $kP_B$ to it. Nobody but A knows the value of $k$, so even though $P_b$ is a public key, nobody can remove the mask $kP_B$. However, A also includes a "clue," which is enough to remove the mask if one knows the private key $n_B$. For an attacker to recover the message, the attacker would have to compute $k$ given $G$ and $kG$, which is assumed to be hard.

As an example of the encryption process (taken from [KOBL94]), take $p = 751$; $E_p(-1, 188)$, which is equivalent to the curve $y^2 = x^3 - x + 188$; and $G = (0, 376)$. Suppose that A wishes to send a message to B that is encoded in the elliptic point $P_m = (562, 201)$ and that A selects the random number $k = 386$. B's public key is $P_{B = (201, 5)}$. We have $386(0, 376) = (676, 558)$, and $(562, 201) + 386(201, 5) = (385, 328)$. Thus, A sends the cipher text $\{(676, 558), (385, 328)\}$.