# Chapter 4

## Public Key Cryptography

**Basic Number theory**

## Groups

A **group** G, sometimes denoted by {G, ·} is a set of elements with a binary operation, denoted by ·, that associates to each ordered pair (a, b) of elements in G an element (a · b) in G, such that the following axioms are obeyed.]The operator · is generic and can refer to addition, multiplication, or some other mathematical operation.

| | |
|---|---|
| (A1) Closure: | If a and b belong to G, then a · b is also in G. |
| (A2) Associative: | a · (b · c) = (a · b) · c for all a, b, c in G. |
| (A3) Identity element: | There is an element e in G such that a · e = e · a = a for all a in G. |
| (A4) Inverse element: | For each a in G there is an element a' in G such that a · a' = a' · a = e. |

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**.

A group is said to be **abelian** if it satisfies the following additional condition:

(A5) Commutative: a · b = b · a for all a, b in G.

*The set of integers (positive, negative, and 0) under addition is an abelian group.*

### Cyclic Group

We define exponentiation within a group as repeated application of the group operator, so that $a^3 = a \cdot a \cdot a$. Further, we define $a^0 = e$, the identity element; and $a^{-n} = (a')^n$. A group G is cyclic if every element of G is a power $a^k$ (k is an integer) of a fixed element a εG. The element a is said to generate the group G, or to be a **generator** of G. A cyclic group is always abelian, and may be finite or infinite.

The additive group of integers is an infinite cyclic group generated by the element 1. In this case, powers are interpreted additively, so that n is the nth power of 1.

## Ring

A **ring** R, sometimes denoted by {R, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed:

(A1-A5) R is an abelian group with respect to addition; that is, R satisfies axioms A1 through A5. For the case of an additive group, we denote the identity element as 0 and the inverse of a as a.

| | |
|---|---|
| (M1) Closure under multiplication: | If a and b belong to R, then ab is also in R. |
| (M2) Associativity of multiplication: | a(bc) = (ab)c for all a, b, c in R. |

(M3) Distributive laws: a(b + c) = ab + ac for all a, b, c in R.
(a + b)c = ac + bc for all a, b, c in R.

***With respect to addition and multiplication, the set of all n-square matrices over the real numbers is a ring.***

### *Commutative Ring*

A ring is said to be **commutative** if it satisfies the following additional condition:

(M4) Commutativity of multiplication: ab = ba for all a, b in R.

Let S be the set of even integers (positive, negative, and 0) under the usual operations of addition and multiplication. S is a commutative ring. The set of all n-square matrices defined in the preceding example is not a commutative ring.

### Integral Domain

we define an **integral domain**, which is a commutative ring that obeys the following axioms:

(M5) Multiplicative identity: There is an element 1 in R such that a1 = 1a = a for all a in R.

(M6) No zero divisors: If a, b in R and ab = 0, then either a = 0 or b = 0.

***Let S be the set of integers, positive, negative, and 0, under the usual operations of addition and multiplication. S is an integral domain.***

### Fields

A **field** F, sometimes denoted by {F, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed:

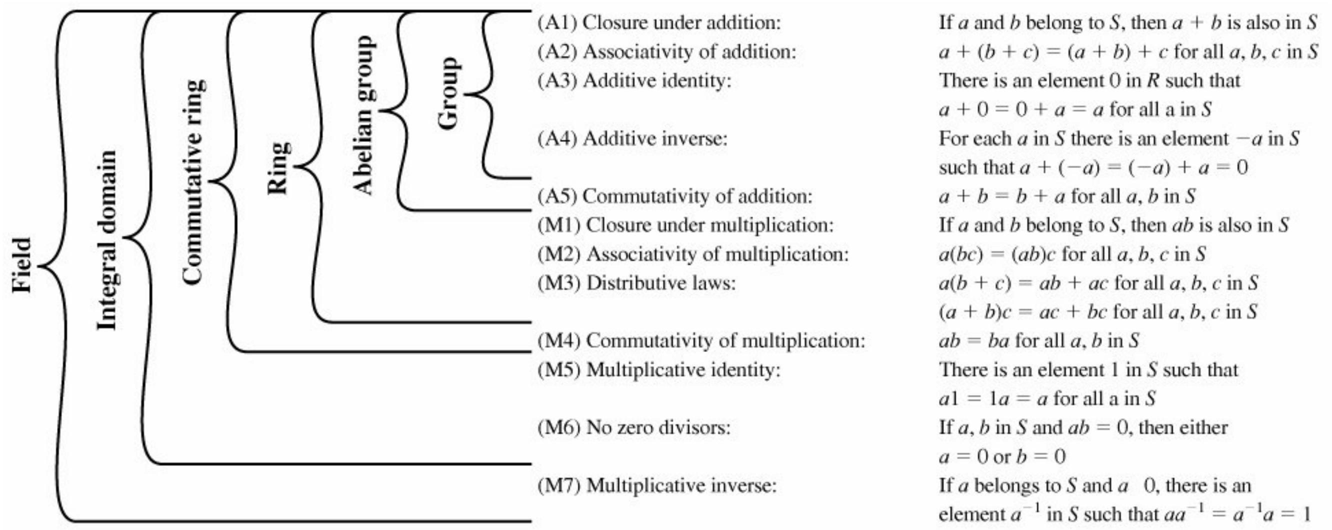
(A1M6) F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6.

(M7) Multiplicative inverse: For each a in F, except 0, there is an element $a^{-1}$ in F such that

$aa^{-1} = (a^{-1})a = 1$.

*In essence, a field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set.* Division is defined with the following rule: $a/b = a(b^{-1})$.

*Familiar examples of fields are the* ***rational numbers****, the* ***real numbers,*** *and the* ***complex numbers****.* Note that the set of **all integers** is not a field, because not every element of the set has a multiplicative inverse; in fact, only the elements 1 and -1 have multiplicative inverses in the integers.



| | |
|---|---|
| (A1) Closure under addition: | If $a$ and $b$ belong to $S$, then $a + b$ is also in $S$ |
| (A2) Associativity of addition: | $a + (b + c) = (a + b) + c$ for all $a, b, c$ in $S$ |
| (A3) Additive identity: | There is an element 0 in $R$ such that $a + 0 = 0 + a = a$ for all a in $S$ |
| (A4) Additive inverse: | For each $a$ in $S$ there is an element $-a$ in $S$ such that $a + (-a) = (-a) + a = 0$ |
| (A5) Commutativity of addition: | $a + b = b + a$ for all $a, b$ in $S$ |
| (M1) Closure under multiplication: | If $a$ and $b$ belong to $S$, then $ab$ is also in $S$ |
| (M2) Associativity of multiplication: | $a(bc) = (ab)c$ for all $a, b, c$ in $S$ |
| (M3) Distributive laws: | $a(b + c) = ab + ac$ for all $a, b, c$ in $S$<br>$(a + b)c = ac + bc$ for all $a, b, c$ in $S$ |
| (M4) Commutativity of multiplication: | $ab = ba$ for all $a, b$ in $S$ |
| (M5) Multiplicative identity: | There is an element 1 in $S$ such that $a1 = 1a = a$ for all a in $S$ |
| (M6) No zero divisors: | If $a, b$ in $S$ and $ab = 0$, then either $a = 0$ or $b = 0$ |
| (M7) Multiplicative inverse: | If $a$ belongs to $S$ and $a$  0, there is an element $a^{-1}$ in $S$ such that $aa^{-1} = a^{-1}a = 1$ |

## Modular Arithmetic

Given any positive integer n and any nonnegative integer a, if **we divide a by n,** we get an integer quotient **q** and an integer remainder **r** that obey the following relationship:

a=qn+r……………………*(1)

If a is an integer and n is a positive integer, we define **a mod n** to be the remainder when a is divided by n. The integer n is called the **modulus**.

**For example: 11 mod 7=4 and -11 mod 7 = 3**

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

**i.e.** $\boldsymbol{a \equiv b \ (mod\ n)}$

## Divisors

We say that a nonzero b divides a if a = mb for some m, where a, b, and m are integers. That is, b divides a if there is no remainder on division. The notation is commonly used to mean b divides a. Also, if b|a, we say that b is a divisor of a.

**The following relations hold:**

- If a|1, then a = ±1.
- If a|b and b|a, then a = ±b.
- Any b ≠0 divides 0.
- If b|g and b|h, then b|(mg + nh) for arbitrary integers m and n.

## Properties of Congruences

Congruences have the following properties:

1. a ≡b (mod n) if n|(a-b).
2. a ≡b (mod n) implies b ≡a (mod n)..
3. a ≡b (mod n) and b ≡c (mod n) imply a ≡c (mod n).

To demonstrate the first point, if n|(a b), then (a b) = kn for some k. So we can write a = b + kn. Therefore, (a mod n) = (reminder when b + kn is divided by n) = (reminder when b is divided by n) = (b mod n)

## Modular Arithmetic Operations

Modular arithmetic exhibits the following properties:

1. [(a mod n) + (b mod n)] mod n = (a + b) mod n
2. [(a mod n) - (b mod n)] mod n = (a- b) mod n
3. [(a mod n) x (b mod n)] mod n = (a x b) mod n

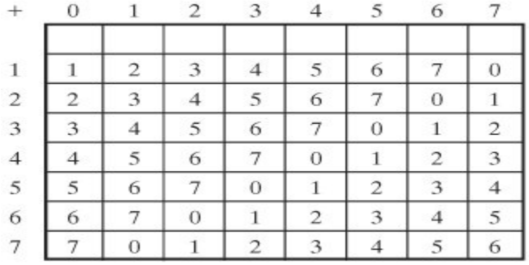
Examples

| **11 mod 8 = 3; 15 mod 8 = 7** |
|---|
| **[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2**<br>**(11 + 15) mod 8 = 26 mod 8 = 2** |
| **[(11 mod 8) (15 mod 8)] mod 8 = 4 mod 8 = 4**<br>**(11 15) mod 8 = 4 mod 8 = 4** |
| **[(11 mod 8) x (15 mod 8)] mod 8 = 21 mod 8 = 5**<br>**(11 x 15) mod 8 = 165 mod 8 = 5** |

Exponentiation is performed by repeated multiplication, as in ordinary arithmetic.

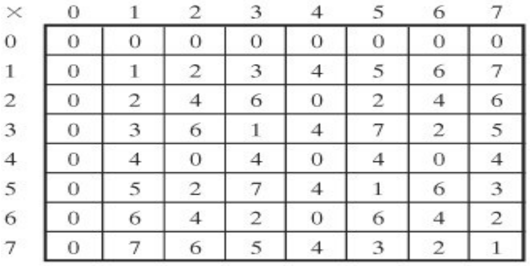
| |
|---|
| To find $11^7$ mod 13, we can proceed as follows: |
| $11^2$ = 121≡ 4 (mod 13) |
| $11^4 = (11^2)^2$ ≡$4^2$ ≡3 (mod 13) |
| $11^7$ ≡11 x 4 x 3≡ 132≡ 2 (mod 13) |

## Arithmetic Modulo 8



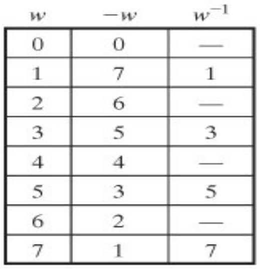
| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 |
| 3 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 |
| 4 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| 5 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 |
| 6 |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



| $w$ | $-w$ | $w^{-1}$ |
|---|---|---|
| 0 | 0 | — |
| 1 | 7 | 1 |
| 2 | 6 | — |
| 3 | 5 | 3 |
| 4 | 4 | — |
| 5 | 3 | 5 |
| 6 | 2 | — |
| 7 | 1 | 7 |

(c) Additive and multiplicative inverses modulo 8

Here –w is additive inverse of w and $w^{-1}$ is the multiplicative inverse of w.

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

$Z_n$={0,1,2,3,…….n-1}.

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

[r] = {a: a is an integer, a≡ r (mod n)}

The residue classes *modulo 4* are

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

If we perform modular arithmetic within $Z_n$, the properties shown in Table (below) hold for integers in $Z_n$. Thus, *$Z_n$ is a commutative ring with a multiplicative identity element.*

|  |  |
|---|---|
| Commutative laws | (w + x) mod n = (x + w) mod n<br>(w x x) mod n = (x x w) mod n |
| Associative laws | [(w + x) + y] mod n = [w + (x + y)] mod n<br>[(w x x) x y] mod n = [w x (x x y)] mod n |

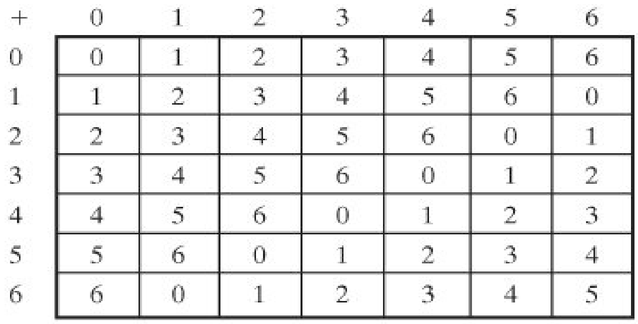
| | |
|---|---|
| Distributive laws | [w + (x + y)] mod n = [(w x x) + (w x y)] mod n<br>[w + (x x y)] mod n = [(w + x) x (w + y)] mod n |
| Identities | (0 + w) mod n = w mod n<br>(1 + w) mod n = w mod n |
| Additive inverse (-w) | For each w $Z_n$, there exists a z such that w + z ≡0 mod n |

*Proof that $Z_8$ is a ring*

## Finite Fields of Order p

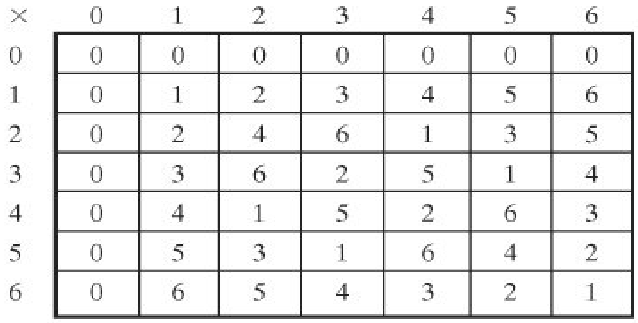
For a given prime, p, the finite field of order p, GF(p) is defined as the set $Z_p$ of integers {0, 1,..., p- 1}, together with the arithmetic operations modulo p. (GF stands for Galois field)

### Arithmetic in GF(7)



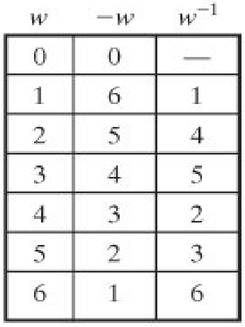
| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |
| 2 | 2 | 3 | 4 | 5 | 6 | 0 | 1 |
| 3 | 3 | 4 | 5 | 6 | 0 | 1 | 2 |
| 4 | 4 | 5 | 6 | 0 | 1 | 2 | 3 |
| 5 | 5 | 6 | 0 | 1 | 2 | 3 | 4 |
| 6 | 6 | 0 | 1 | 2 | 3 | 4 |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

(a) Addition modulo 7



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

(b) Multiplication modulo 7



| $w$ | $-w$ | $w^{-1}$ |
|---|---|---|
| 0 | 0 | — |
| 1 | 6 | 1 |
| 2 | 5 | 4 |
| 3 | 4 | 5 |
| 4 | 3 | 2 |
| 5 | 2 | 3 |
| 6 | 1 | 6 |

(c) Additive and multiplicative inverses modulo 7

### Finding the Multiplicative Inverse in GF(p)

It is easy to find the multiplicative inverse of an element in GF(p) for small values of p. You simply construct a multiplication table, such as shown in Table above, and the desired result can be read directly. However, for large values of p, this approach is not practical.

If gcd(m, b) = 1, then b has a multiplicative inverse modulo m. That is, for positive integer b < m, there exists a b1 < m such that bb1 = 1 mod m. The Euclidean algorithm can be extended so that, in addition to finding gcd(m, b), if the gcd is 1, the algorithm returns the multiplicative inverse of b.

```
EXTENDED EUCLID(m, b)
1. (A1, A2, A3) ←(1, 0, m); (B1, B2, B3) ←(0, 1, b)
2. if B3 = 0 return A3 = gcd(m, b); no inverse
3. if B3 = 1 return B3 = gcd(m, b); B2 = $b^1$ mod m


4. Q=$\left\lfloor \frac{A3}{B3} \right\rfloor$

5. (T1, T2, T3) ← (A1  QB1, A2  QB2, A3  QB3)
6. (A1, A2, A3) ← (B1, B2, B3)
7. (B1, B2, B3) ← (T1, T2, T3)
8. goto 2
```

Exercise: trace the above algorithms for finding multiplicative inverse of 550 in GF(1759)

## Prime Numbers

### Prime Numbers

An integer p > 1 is a prime number if and only if its only divisors are ± 1 and ±p. Examples are 7 , 13…

Any integer a > 1 can be factored in a unique way as:
$A={p1}^{a1}.{p2}^{a2}........................{pt}^{at}$ where p1 < p2 < ... < pt are prime numbers and where each is a positive integer. This is known as *the fundamental theorem of arithmetic*. Examples are

91 = 7 x 13 (factorization)
3600 = 24 x 32 x 52
11011 = 7 x 112 x 13

### Fermat’s theorem

Fermat's theorem states the following: If p is prime and a is a positive integer not divisible by p, then
$a^{p-1}$=1 (mod p)
(here a and p are relatively prime)

### Example

| |
|---|
| a = 7, p = 19 |
| $7^2$ = 49 ≡11(mod 19) |
| $7^4$ =$7^2$ x$7^2$≡11x11=121 ≡7(mod 19) |
| $7^8$≡ 49 ≡11(mod 19) |
| $7^{16}$ =121 =7(mod 19) |

| |
|---|
| $a^{p1} = 7^{18} = 7^{16}$ x $7^2$= 7 x 11 =1(mod 19) |

Alternative form of fermat theorem is $a^p$=a(mod p)

**Euler's Totient Function**

Before presenting Euler's theorem, we need to introduce an important quantity in number theory, referred to as Euler's totient function and written ϕ(n), defined as the number of positive integers less than n and relatively prime to n. By convention, ϕ(1) = 1.

| **n** | **ϕ(n)** |
|---|---|
| 1 | 1 |
| 3 | 2 |
| 13 | 12 |
| 14 | 6 |
| 15 | 8 |
| 19 | 18 |
| 20 | 8 |

**If n is prime number then** ϕ(n)=n-1.

**Euler's theorem**

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

$$a^{\emptyset(n)} = 1(mod\ n)$$

**Examples**

| | |
|---|---|
| a = 3; n = 10; ϕ(10) = 4 | $a^{\phi(n)} = 3^4$ = 81 ≡1(mod 10) = 1 (mod n) |
| a = 2; n = 11; ϕ(11) = 10 | $a^{\phi(n)} = 2^{10}$ = 1024 ≡1(mod 11) = 1 (mod n) |

***An alternative form of Euler's theorem is***

$$a^{\emptyset(n)+1} = a(mod\ n)$$

## Testing for Primality

For many cryptographic algorithms, it is necessary to select one or more very large prime numbers at random. Thus we are faced with the task of determining whether a given large number is prime. There is no simple yet efficient means of accomplishing this task.

## Miller-Rabin Algorithm

The algorithm due to Miller and Rabin is typically used to test a large number for primality. Before explaining the algorithm, we need some background.

First, any positive odd integer n>= 3 can be expressed as follows:

$n - 1 = 2^k q$ with $k > 0$, q odd

### Two Properties of Prime Numbers

### The first property is stated as follows:

If p is prime and a is a positive integer less than p, then $a^2 \bmod p = 1$ if and only if either a mod p = 1 or a mod p = p -1. By the rules of modular arithmetic (a mod p) (a mod p) = $a^2$ mod p. Thus if either a mode p = 1 or a mod p = p -1, then $a^2$ mod p = 1.

### The second property is stated as follows:

Let p be a prime number greater than 2. We can then write $p - 1 = 2^k q$, with k > 0 q odd. Let a be any integer in the range 1 < a < p - 1. Then one of the two following conditions is true:

1. $a^q$ is congruent to 1 modulo p. That is, $a^q \bmod p = 1$, or equivalently, $a^q \equiv 1 \pmod p$.
2. One of the numbers $a^q, a^{2q}, a^{4q}, \ldots, a^{2^{k-1}}q$ is congruent to 1 modulo p. That is, there is some number j in the range (1 <=j <<k) such that $a^{2j-1q}$ mod p = 1 mod p = p-1, or equivalently, $a^{2j-1q} \equiv 1 \pmod p$.

### Details of Miller Rubin algorithm

These considerations lead to the conclusion that if n is prime, then either the first element in the list of residues, or remainders, (aq, $a^2$q,..., , $a^{2^{k-1}}$ q, , $a^{2^k}$q) modulo n equals 1, or some element in the list equals (n-1); otherwise n is composite (i.e., not a prime). On the other hand, if the condition is met, that does not necessarily mean that n is prime.

For example, if n = 2047 = 23 x 89, then n 1 = 2 x 1023. Computing, 21023 mod 2047 = 1, so that 2047 meets the condition but is not prime.

We can use the preceding property to devise a test for primality. The procedure TEST takes a candidate integer n as input and returns the result `composite` if n is definitely not a prime, and the result `inconclusive` if n may or may not be a prime.

```
TEST (n)
```

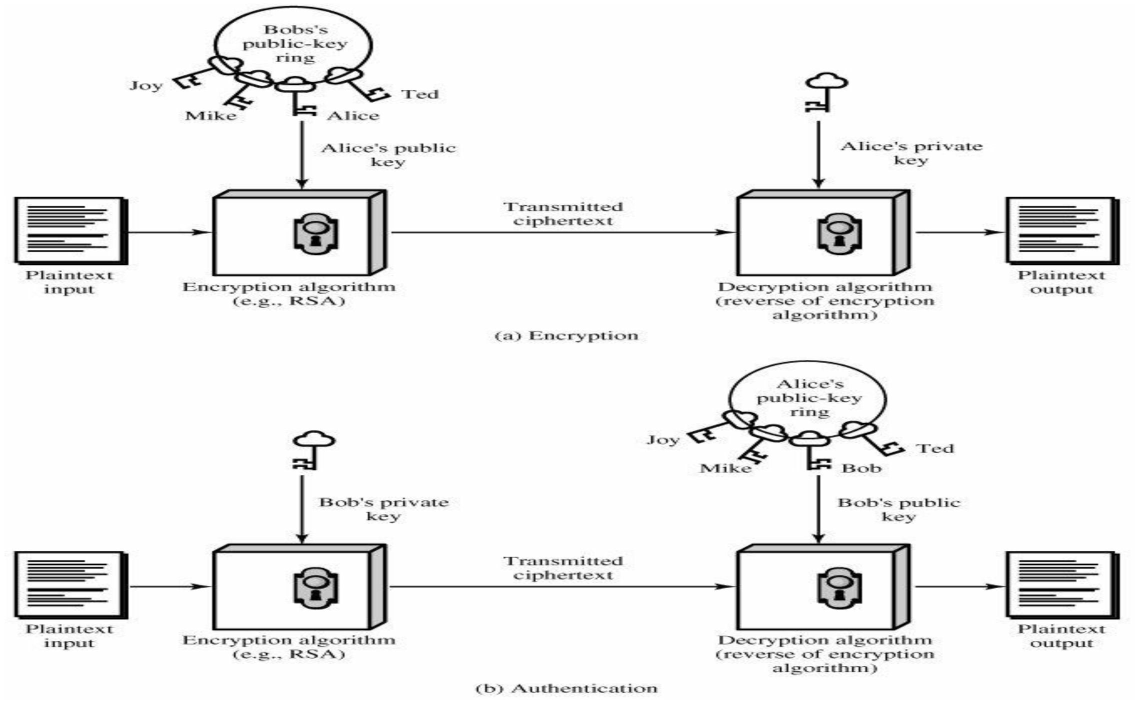
```
Find integers k, q, with k > 0, q odd, so that (n-1 = $2^k q$);
Select a random integer a, 1 < a < n-1;
if $a^q$ mod n = 1 then return("inconclusive");
for j = 0 to k  1 do
   if $a^{2jq}$ mod n ≡ n  1 then return("inconclusive");
return("composite");
```

### Public key Cryptography: RSA

## 1. Public-Key Cryptosystems

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 and private keys:** 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. **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.
5. **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.



### 1.1 Public-Key Cryptosystem for Secrecy

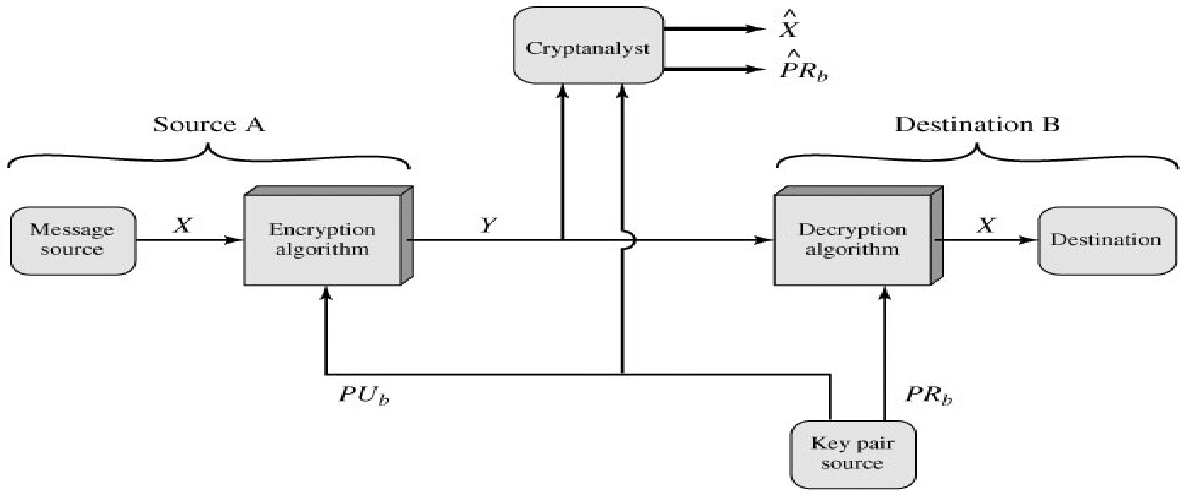
There is some source A that produces a message in plaintext, X =[$X_1$, $X_2$,..., $X_M$,]. The M elements of X are letters in some finite alphabet. The message is intended for destination B. B generates a related pair of keys: a public key, $PU_b$, and a private key, $PR_b$. $PR_b$ is known only to B, whereas $PU_b$ is publicly available and therefore accessible by A. this is shown in figure next page.

With the message X and the encryption key $PU_b$ as input, A forms the ciphertext Y = [$Y_1$, $Y_2$,..., $Y_N$]:

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

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

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

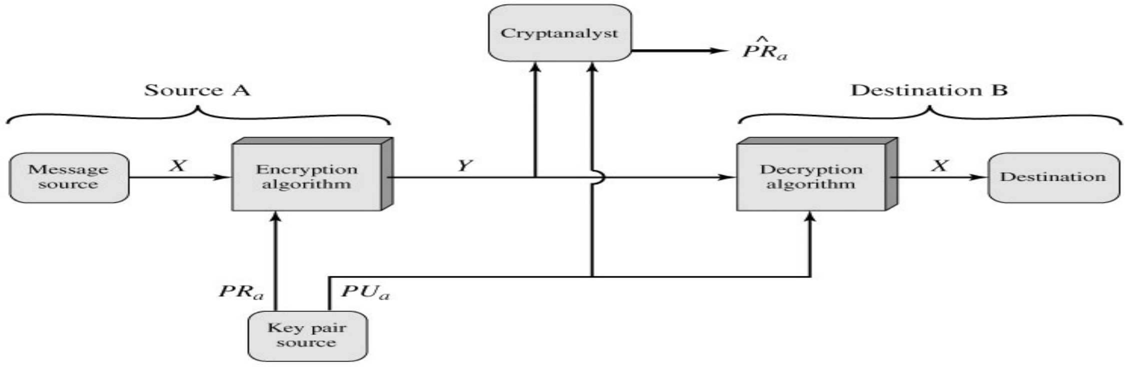


## 1.2 Public-Key Cryptosystem: Authentication

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. In addition, 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. Figure show the use of public-key encryption to provide authentication:

$Y = E(PR_a, X)$

$Y = E(PU_a, Y)$



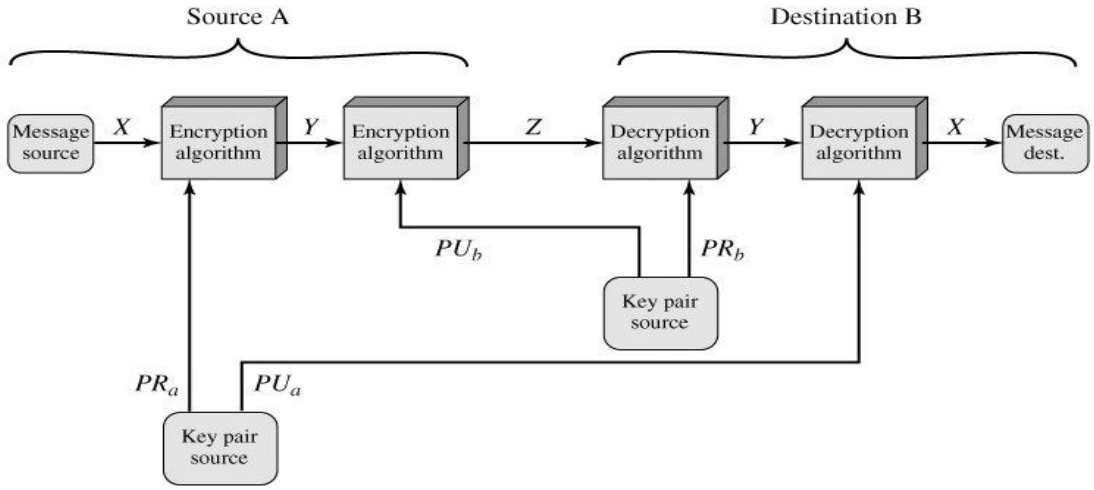
## 1.3 Public-Key Cryptosystem: Authentication and Secrecy

It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme:

$Z = E(PU_b, E(PR_a, X))$

$X = D(PU_a, E(PR_b, Z))$

In this case, we begin as before by encrypting a message, using the sender's private key. This provides the digital signature. Next, we encrypt again, using the receiver's public key. The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus, confidentiality is provided. The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.



**Requirements for Public-Key Cryptography**

1. It is computationally easy for a party B to generate a pair (public key $PU_b$, private key $PR_b$).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:

   $C = E(PU_b, M)$

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

   $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$

4. It is computationally infeasible for an adversary, knowing the public key, $PU_b$, to determine the private key, $PR_b$.
5. It is computationally infeasible for an adversary, knowing the public key, $PU_b$, and a ciphertext, C, to recover the original message, M.

   We can add a sixth requirement that, although useful, is not necessary for all public-key applications:

6. The two keys can be applied in either order:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

**There are three families of Public-Key (PK) algorithms of practical relevance:**
1. Integer factorization algorithms (RSA, ...)
2. Discrete logarithms (Diffie-Hellman, DSA, ...)
3. Elliptic curves (EC)

In this lecture we only consider Algorithms of family i.e. Integer Factorization Algorithm and Discrete Logarithms.

## 1. The RSA Algorithm

RSA is an algorithm for public-key cryptography. It was the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.

**Operation:** RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key.

**RSA Key Generation:**

1. Choose two distinct large random prime numbers p and q
2. Compute n = pq, n is used as the modulus for both the public and private keys
3. Compute the totient: $\varphi(n) = (p - 1)(q - 1)$.
4. Choose an integer e such that $1 < e < \varphi(n)$, and e and $\varphi(n)$ share no factors other than 1 i.e. e and $\varphi(n)$ are relatively prime)
5. e is released as the public key exponent
6. Compute d to satisfy the congruence relation $ed \equiv 1 \bmod \varphi(n)$; i.e. $de = 1 + k\varphi(n)$ for some integer k.
7. d is kept as the private key exponent

**Notes on the above steps:** Step 1: Numbers can be probabilistically tested for primality.

Step 4: A popular choice for the public exponents is e = $2^{16} + 1 = 65537$. Some applications choose smaller values such as e = 3, 5, 17 or 257 instead. This is done to make encryption and signature verification faster.

Steps 4 and 5 can be performed with the extended Euclidean algorithm;

### Encrypting Messages

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob send message M to Alice by turning M into a number m < n by using a reversible protocol called a padding scheme. He then computes the ciphertext c as: $c = m^e$ **mod** n. Bob then transmits c to Alice.

### Decrypting Messages

Alice can recover m from c by using her private key exponent d by the following computation:

$m = c^d$ **mod** n. Given m, she can recover the original message M.

**Example:** Consider, p = 61 and q = 53 now, compute n = pq = 61 * 53 = 3233

Compute the totient φ(n) = (p − 1)(q − 1) = (61-1)(53-1) = 3120

Choose e > 1 relatively prime to 3120; e = 17

Compute d such that ed ≡ 1 mod φ(n) e.g., by computing the modular multiplicative inverse of e modulo φ(n): d = 2753 since 17 * 2753 = 46801 = 1 + 15 * 3120.

The public key is (n = 3233, e = 17).

For a padded message m the encryption function is:

$$c = m^e \textbf{ mod } n = m^{17} \textbf{ mod } 3233.$$

The private key is (n = 3233, d = 2753). The decryption function is:

$$m = c^d \textbf{ mod } n = c^{2753} \textbf{ mod } 3233.$$

For example, to encrypt m = 123, we calculate

$c = 123^{17}$ **mod** 3233 = 855 to decrypt c = 855, we calculate $m = 855^{2753}$ **mod** 3233 = 123

Both of these calculations can be computed efficiently using the square-and-multiply algorithm for modular exponentiation.

**One More Example:**

Consider primes p=11, q=3. Now, compute n = pq = 11.3 = 33 and

totient φ(n) = (p-1)(q-1) = 10.2 = 20 **.**

Choose e=3; Check gcd(e, φ(n)) = gcd(3, 20) = 1 (i.e. 3 and 20 have no common factors except 1),

Compute d such that ed ≡ 1 (mod φ(n) )
i.e. find a value for d such that φ(n) divides (ed-1)
i.e. find d such that 20 divides 3d-1.
Simple testing (d = 1, 2, ...) gives d = 7
Check: ed-1 = 3.7 - 1 = 20, which is divisible by φ(n)

**The notation 'a ≡ b (mod n)' means a and b have the same remainder when divided by n, or, equivalently,**

Public key = (n, e) = (33, 3)
Private key = (n, d) = (33, 7).

This is actually the smallest possible value for the modulus n for which the RSA algorithm works. Now say we want to encrypt the message m = 7,
c = $m^e$ mod n = $7^3$ mod 33 = 343 mod 33 = 13.
Hence the ciphertext c = 13. To check decryption we compute m' = $c^d$ mod n = $13^7$ mod 33 = 7.

## 2. Diffie-Hellman Key Exchange

Diffie-Hellman (D-H) key exchange is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. Other names for Diffie-Hellman Key Exhange are *Diffie-Hellman Key Agreement, Diffie-Hellman Key Establishment, Diffie-Hellman Key Negotiation, Exponential Key Exchange.*

**Description:** The simplest and original implementation of the protocol uses the multiplicative group of integers modulo p, where p is a prime and g is primitive root of p.

**Steps:**

1. Generate the global public elements p and g, where p is a prime number and g < p is a primitive root of p.
2. User A selects a random integer number $X_A$<p, and computes $Y_A = g^{X_A}$ mod p.
3. User B independently selects a random integer $X_B$<p, and computes

   $Y_B = g^{X_B}$ mod p.
4. Each side keeps the X value private and makes the Y value available publicly to the other side.
5. User A generates secret key as K = $(Y_B)^{X_A}$ mod p.

6. User B generates secret key as $K = (Y_A)^{X_B} \bmod p$

**Why the key from both side same:**

From user A, $K = (Y_B)^{X_A} \bmod p = (g^{X_B} \bmod p)^{X_A} \bmod p = (g^{X_B})^{X_A} \bmod p = g^{X_B X_A} \bmod p$

From user B, $K = (Y_A)^{X_B} \bmod p = (g^{X_A} \bmod p)^{X_B} \bmod p = (g^{X_A})^{X_B} \bmod p = g^{X_B X_A} \bmod p$

See above both the results are same.

**Example:** Alice and Bob agree to use a prime number p=23 and base g=5.

Alice chooses a secret integer $X_A = 6$, then sends Bob ($Y_A = g^{X_A} \bmod p$):$5^6 \bmod 23 = 8$.

Bob chooses a secret integer $X_B = 15$, then sends Alice ($Y_B = g^{X_B} \bmod p$):$5^{15} \bmod 23 = 19$.

Alice computes $(Y_B)^{X_A} \bmod p$: $19^6 \bmod 23 = 2$ and Bob computes $(Y_A)^{X_B} \bmod p$: $8^{15} \bmod 23 = 2$.

Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of $X_A$, $X_B$, and p would be needed to make this example secure, since it is easy to try all the possible values of $g^{X_A X_B} \bmod 23$ (there will be, at most, 22 such values, even if $X_A$, $X_B$ are large). If p were a prime of at least 300 digits, and $X_A$, $X_B$ were at least 100 digits long, then even the best algorithms known today could not find a given only g, p, and $g^{X_A} \bmod p$, even using all of mankind's computing power. The problem is known as the discrete logarithm problem. *Note that g need not be large at all, and in practice is usually either 2 or 5.*