LECTURE NOTES

PROGRAMME – BCA VI SEM

CRYPTOGRAPHY

PAPER CODE:- BCA-6044

UNIT III

Public Key Encryption: Public Key Encryption: public,key cryptography: principles of public, key cryptosystems, RSA algorithm, key management, Fermat's & Euler's theorem, primality test and the Chinese remainder theorem.

Public Key Cryptography The development of public-key cryptography is the greatest and perhaps the only true revolution in the entire history of cryptography. It is asymmetric, involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key. Public key schemes are neither more nor less secure than private key (security depends on the key size for both). Public-key cryptography complements rather than replaces symmetric cryptography. Both also have issues with key distribution, requiring the use

of some suitable protocol. The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption: 1.) key distribution – how to have secure communications in general without having to trust a KDC with your key 2.) digital signatures – how to verify a message comes intact from the claimed sender Public-key/two-key/asymmetric cryptography involves the use of two keys: ¬ a public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures ¬ a private-key, known only to the recipient, used to decrypt messages, and sign (create) signatures. ¬ is asymmetric because those who encrypt messages or verify signatures cannot decrypt messages or create signatures Public-Key algorithms rely on one key for encryption and different but related key for decryption. These algorithms have the following important characteristics: ¬ it is computationally infeasible to find decryption key knowing only algorithm & encryption key ¬ it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known ¬ either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms like RSA)

Asymmetric Encryption

Mathematics of Asymmetric Key Cryptography, Asymmetric Key Cryptography

Primes and Related Congruence Equations

PRIMES

Asymmetric-key cryptography uses prime numbers extensively.
A prime is divisible only by itself and 1.

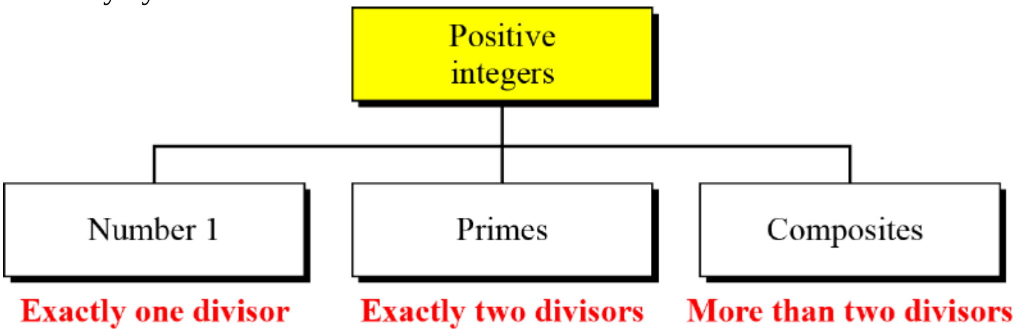


Figure Three groups of positive integers

Example 1:
What is the smallest prime?

The smallest prime is 2, which is divisible by 2 (itself) and 1.

Example 2:

List the primes smaller than 10.

There are four primes less than 10: 2, 3, 5, and 7. It is interesting to note that the percentage of primes in the range 1 to 10 is 40%. The percentage decreases as the range increases.

Cardinality of Primes

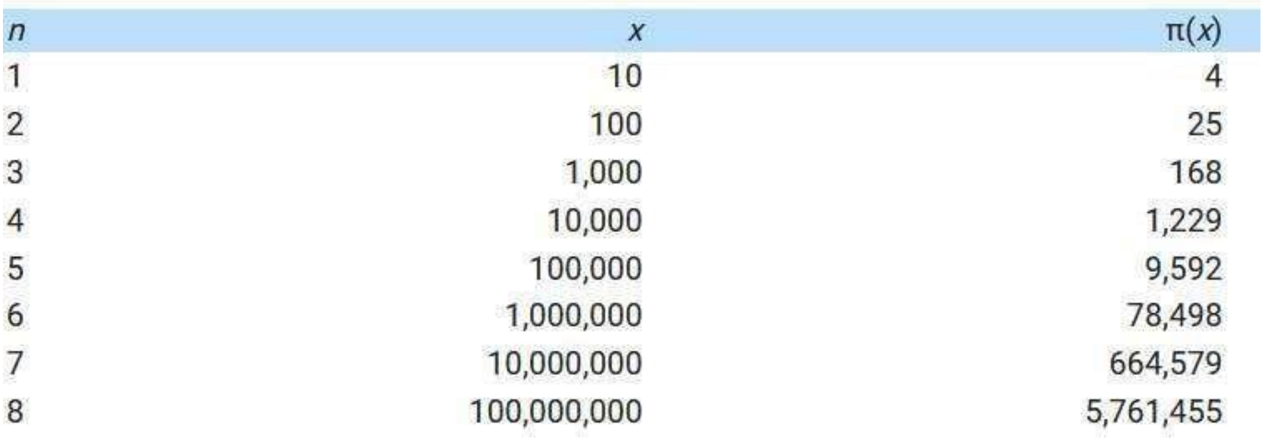
We can use infinite Number of Primes.

Number of Primes

π(x) is the number of primes less than or equal to x. π is not similar to mathematics π.
The primes under 25 are 2, 3, 5, 7, 11, 13, 17, 19 and 23 so π(3) = 2, π(10) = 4 and π(25) = 9.

$$\boxed{[n / (\ln n)] \quad < \quad \pi(n) \quad < \quad [n/(\ln n - 1.08366)]}$$

A Table of values of π(*x*)



n	*x*	π(*x*)
1	10	4
2	100	25
3	1,000	168
4	10,000	1,229
5	100,000	9,592
6	1,000,000	78,498
7	10,000,000	664,579
8	100,000,000	5,761,455

Example 1

Find the number of primes less than 1,000,000.
The approximation gives the range 72,383 to 78,543.

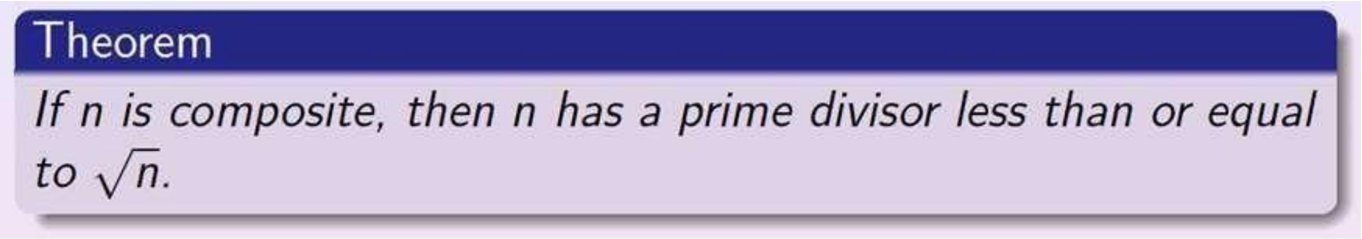
The actual number of primes is 78,498.

Checking for Primeness

Given a number n, how can we determine if n is a prime? The answer is that we need to see if the number is divisible by all primes less than

$$\sqrt{n}$$

We know that this method is inefficient, but it is a good start.



Theorem

If n is composite, then n has a prime divisor less than or equal to $\sqrt{n}$.

Example 1:
Is 97 a prime?

The floor of **π**(97) = 9. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.
Example 2:
Is 301 a prime?

The floor of **π**(301) = 17. We need to check 2, 3, 5, 7, 11, 13, and 17. The numbers 2, 3, and 5 do not divide 301, but 7 does. Therefore 301 is not a prime.

Fermat’s Little Theorem

First Version: if p is prime and a is positive integer, then

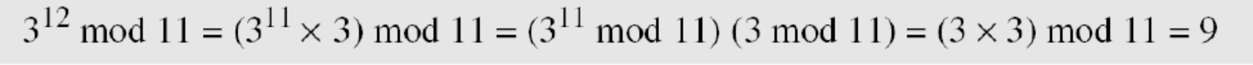
$$a^{p-1} \equiv 1 \bmod p$$

Second Version:

$$a^{p} \equiv a \bmod p$$

This means that if we divide a^p by p then the remainder should be 'a'.

Example 1:
Find the result of 6^{10} mod 11.
We have 6^{10} mod 11 = 1. This is the first version of Fermat's little theorem where $p = 11$.
Example 2
Find the result of 3^{12} mod 11.
Here the exponent (12) and the modulus (11) are not the same. With substitution this can be solved using Fermat's little theorem.

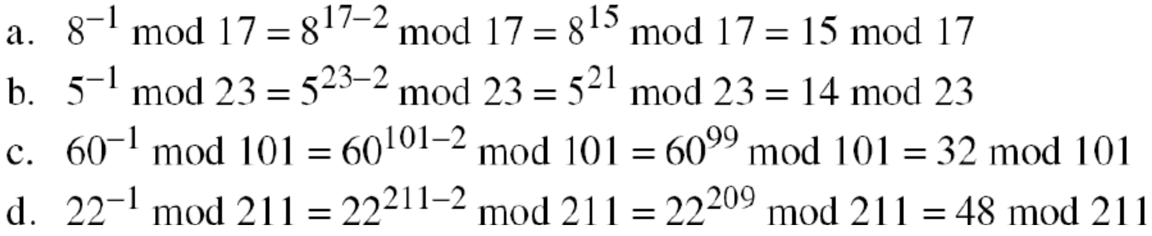


$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11)\ (3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

Multiplicative Inverses

$$a^{-1} \bmod p = a^{p-2} \bmod p$$

Example
The answers to multiplicative inverses modulo a prime can be found without using the extended Euclidean algorithm:



a. $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$

b. $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$

c. $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$

d. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

Example:

How to calculate multiplicative inverse of 5 modulo 23 that is 5^{-1} mod 23?

Solution:

1. 5^{-1} mod 23 = 5^{23-2} mod 23 (Ref: a^{-1} mod p= a^{p-2} mod p)
2. 5^{23-2} mod 23 = 5^{21} mod 23
3. Calculate following to solve 5^{21} mod 23:
   5^1 mod 23 = 5
   5^2 mod 23=25 mod 23=2
   5^4 mod 23= $(5^2)^2$ mod 23= $(2)^2$ mod 23=4
   5^8 mod 23= $(5^4)^2$ mod 23 $(4)^2$ mod 23=16
   5^{16} mod 23= $(5^8)^2$ mod 23 $(16)^2$ mod23=256 mod 23=3

Now binary equivalence of 21 is 10101, so multiply 5^1 , 5^4 and 5^{16} values, leave 5^2 and 5^8 because these are 0's in binary form.

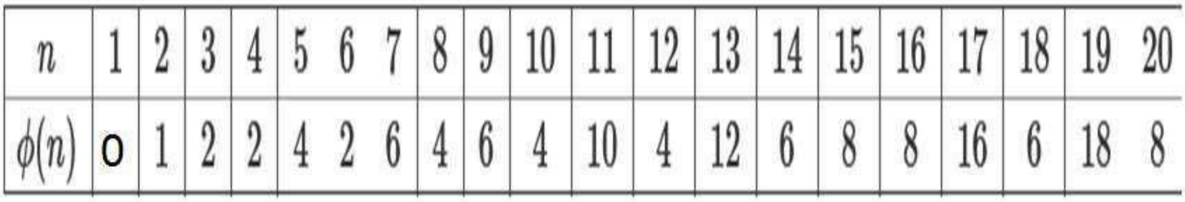
5^{21} mod 23 = (5^{16} x 5^4 x 5^1) mod 23=(3x4x5) mod 23=60 mod 23= 14 mod 23.

Finally 5^{-1} mod 23 = 5^{21} mod 23 = 14 mod 23

Euler's totient function

Euler's totient function, also known as **phi-function** $\phi(n)$, this function counts the number of integers that are both smaller than n and relatively prime to n (coprime). Two numbers are coprime if their greatest common divisor equals 1.

Here are values of $\phi(n)$ for the first few positive integers:



n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\phi(n)$	0	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18	8

Example: Find co-primes of 9?

If we check gcd(9,1), gcd(9,2), gcd(9,4), gcd(9,5), gcd(9,7), gcd(9,8) =1,

So, coprimes to 9 are 1,2,4,5,7,8 and their count $\phi(9)=6$

Properties

- $\phi(1)=0$
- If p is a prime number, $\phi(p)=p-1$
- If a and b are relatively prime, then: $\phi(ab)=\phi(a)\cdot\phi(b)$.
- If p is a prime, $\phi(p^e)=p^e - p^{e-1}$

Examples:

1) Find $\phi(7)$?

$\phi(7)=7-1=6$

2)Find $\phi(21)$?

$\phi(21)= \phi(3x7) = \phi(3)x\ \phi(7)=2x6=12$

3)Find $\phi(77)$?

$\phi(77)= \phi(7x11) = \phi(7)x\ \phi(11)=6x10=60$

4) Find $\phi(3^2)$?

$\phi(3^2)= (3^2)- (3^{2-1}) = 9-3=6$

5) What is the value of ϕ (13)?

Because 13 is a prime, ϕ (13) = (13 −1) = 12.

6)What is the value of ϕ (10)?

We can use the third rule: ϕ (10) = ϕ (2) × ϕ (5) = 1 × 4 = 4, because 2 and 5 are primes.

7)What is the value of ϕ (240)?

We can write $240 = 2^4 \times 3^1 \times 5^1$. Then

$$\phi\ (240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$$

8)Can we say that ϕ (49) = ϕ (7) × ϕ (7) = 6 × 6 = 36?

No. The third rule applies when m and n are relatively prime. Here $49 = 7^2$. We need to use the fourth rule: ϕ (49) = $7^2 - 7^1 = 42$.

9)What is the number of elements in Z_{14}*?

The answer is ϕ (14) = ϕ (7) × ϕ (2) = 6 × 1 = 6. The members are 1, 3, 5, 9, 11, and 13.

Note: Interesting point: If $n > 2$, the value of f(n) is even.

Euler’s Theorem

First Version:For every a and n, they are relatively prime then

$$a^{\varphi(n)} \equiv 1 \pmod n$$

Second Version

$$a^{k \times f(n) + 1} \equiv a \pmod n$$

Note: The second version of Euler’s theorem is used in the RSA cryptosystem.

Example: if a=3 and n=10, show that $3^{\phi(10)} \equiv 1 \bmod 10$

$\phi(10)=\phi(2) x \phi(5)=1x4=4$

$3^{\phi(10)}=3^4=81$

$3^{\phi(10)} \equiv 1 \bmod 10$

$81 \equiv 1 \bmod 10$ is true because $81 \bmod 10 = 1$

Example 2:
Find the result of 6^{24} mod 35.
Solution
We have 6^{24} mod 35 = $6^{\phi\ (35)}$ mod 35 = 1.
Example :
Find 3^4 mod 10 ?
Solution

We have $3^4 = 3^{\phi(10)} \bmod 10 = 1$ because $\phi(10)= \phi(2) x\ \phi(5)=1x4=4$

Example 3:
Find the result of 20^{62} mod 77.
Solution
If we let k = 1 on the second version,
we have f(77)= f(7)x f(11)=6x10=60
20^{62} mod 77 = (20 mod 77) (20^{60+1} mod 77) mod 77=
(20 mod 77) ($20^{f(77)\ +\ 1}$ mod 77) mod 77
= (20)(20) mod 77 = 15.

Multiplicative Inverses

Euler's theorem can be used to find multiplicative inverses modulo a composite.

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

Example:
The answers to multiplicative inverses modulo a composite can be found without using the extended Euclidean algorithm if we know the factorization of the composite:

a. $8^{-1} \bmod 77 = 8^{\phi(77)-1} \bmod 77 = 8^{59} \bmod 77 = 29 \bmod 77$
b. $7^{-1} \bmod 15 = 7^{\phi(15)-1} \bmod 15 = 7^{7} \bmod 15 = 13 \bmod 15$
c. $60^{-1} \bmod 187 = 60^{\phi(187)-1} \bmod 187 = 60^{159} \bmod 187 = 53 \bmod 187$
d. $71^{-1} \bmod 100 = 71^{\phi(100)-1} \bmod 100 = 71^{39} \bmod 100 = 31 \bmod 100$

Primitive Root and Multiplicative Orders

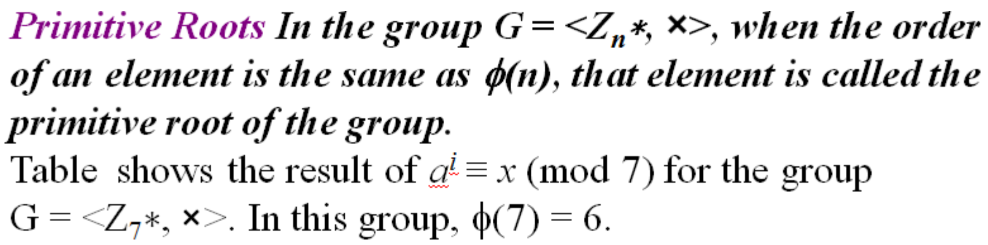
Multiplicative Order:

If 'a' and 'n' are relatively prime, then
The multiplicative order of 'a' modulo n is smallest positive integer 'k' with
$a^k \equiv 1$ (mod n)

The order of modulo 'n' is written as $ord_n(a)$ or $O_n(a)$
Example 1: Define multiplicative order of 4 mod 7

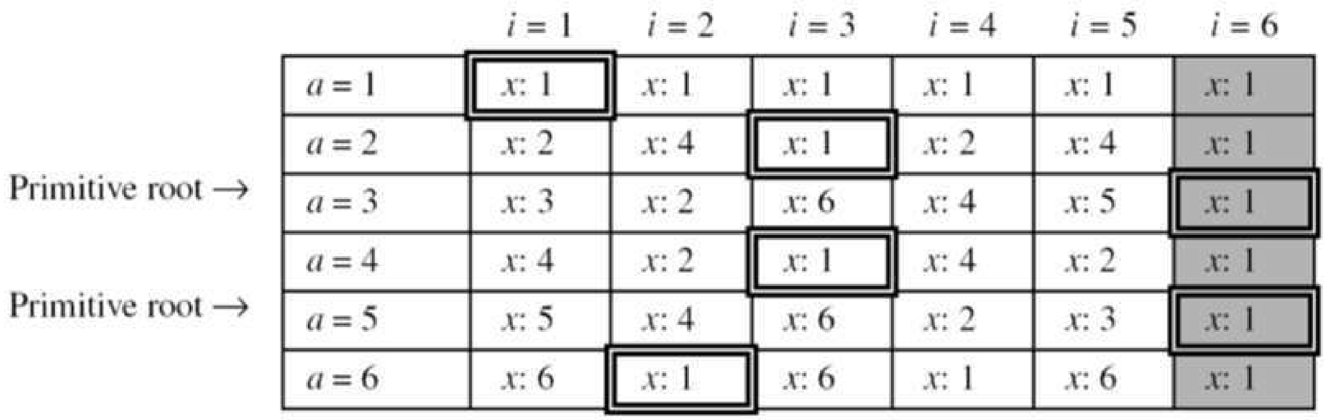
$4^1=4 \equiv 3 \pmod 7$
$4^2=16 \equiv 2 \pmod 7$
$4^3=64 \equiv 1 \pmod 7$
$\mathrm{Ord}_7(4)=3$ because 4^3 is congruent to 1 modulo 7.
Example 2: Define multiplicative order of 2 mod 7
$2^1=2 \equiv 2 \pmod 7$
$2^2=4 \equiv 4 \pmod 7$
$2^3=8 \equiv 1 \pmod 7$
$\mathrm{Ord}_7(2)=3$ because 2^3 is congruent to 1 modulo 7.

Primitive Root :

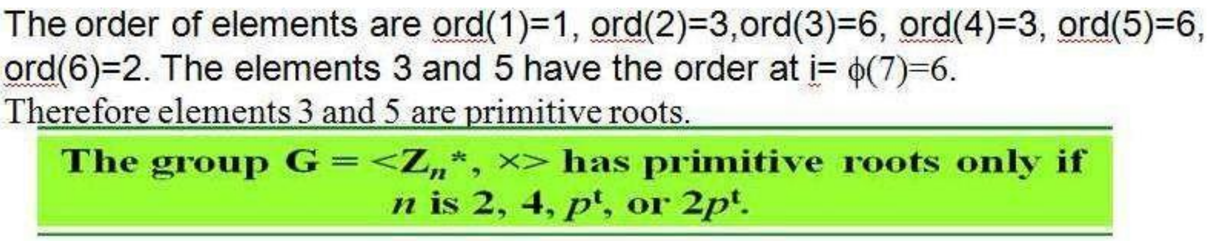


Primitive Roots In the group $G = \langle Z_n*, \times\rangle$, when the order of an element is the same as $\phi(n)$, that element is called the primitive root of the group.

Table shows the result of $a^i \equiv x \pmod 7$ for the group $G = \langle Z_7*, \times\rangle$. In this group, $\phi(7) = 6$.



		$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
	$a=1$	x: 1	x: 1	x: 1	x: 1	x: 1	x: 1
	$a=2$	x: 2	x: 4	x: 1	x: 2	x: 4	x: 1
Primitive root →	$a=3$	x: 3	x: 2	x: 6	x: 4	x: 5	x: 1
	$a=4$	x: 4	x: 2	x: 1	x: 4	x: 2	x: 1
Primitive root →	$a=5$	x: 5	x: 4	x: 6	x: 2	x: 3	x: 1
	$a=6$	x: 6	x: 1	x: 6	x: 1	x: 6	x: 1



The order of elements are ord(1)=1, ord(2)=3,ord(3)=6, ord(4)=3, ord(5)=6, ord(6)=2. The elements 3 and 5 have the order at i= ϕ(7)=6.
Therefore elements 3 and 5 are primitive roots.

The group G = <Z_n*, ×> has primitive roots only if n is 2, 4, p^t, or $2p^t$.

If the Group G=<Z_n^*,x> has any primitive root, the number of primitive roots is

φ(φ (n))

Example: Find the Number of primitive roots of 25
ϕ (25)=20

Find the primitive root of 761
ϕ (ϕ (761))= ϕ (760)

= ϕ (2^3x5x19) = ϕ (2^3)x ϕ (5)x ϕ (19)

=(2^3 - 2^2)x 4x18=4x4x18
=288

CHINESE REMAINDER THEOREM

The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$\begin{gathered} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \cdots \\ x \equiv a_k \pmod{m_k} \end{gathered}$$

Solution To Chinese Remainder Theorem

1. Find M = $m_1 \times m_2 \times \ldots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1$, $M_2 = M/m_2$, …, $M_k = M/m_k$.
3. Find the multiplicative inverse of M_1, M_2, …, M_k using the corresponding moduli (m_1, m_2, …, m_k). Call the inverses M_1^{-1}, M_2^{-1}, …, M_k^{-1}.
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

Example:

Find the solution to the simultaneous equations:

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{7}$$

Solution:
We follow the four steps.

1. $M = 3 \times 5 \times 7 = 105$
2. $M_1 = 105 / 3 = 35$, $M_2 = 105 / 5 = 21$, $M_3 = 105 / 7 = 15$
3. The inverses are $M_1^{-1} = 2$, $M_2^{-1} = 1$, $M_3^{-1} = 1$
4. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105 = 23 \bmod 105$

Example 2:
Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.
Solution
This is a CRT problem. We can form three equations and solve them to find the value of x.

$$x = 3 \bmod 7$$
$$x = 3 \bmod 13$$
$$x = 0 \bmod 12$$

If we follow the four steps, we find x = 276. We can check that
276 = 3 mod 7, 276 = 3 mod 13 and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

Example 3
Assume we need to calculate $z = x + y$ where $x = 123$ and $y = 334$, but our system accepts only numbers less than 100.

$$x \equiv 24 \pmod{99} \qquad y \equiv 37 \pmod{99}$$
$$x \equiv 25 \pmod{98} \qquad y \equiv 40 \pmod{98}$$
$$x \equiv 26 \pmod{97} \qquad y \equiv 43 \pmod{97}$$

Adding each congruence in x with the corresponding congruence in y gives

$$x + y \equiv 61 \pmod{99} \quad \rightarrow \quad z \equiv 61 \pmod{99}$$
$$x + y \equiv 65 \pmod{98} \quad \rightarrow \quad z \equiv 65 \pmod{98}$$
$$x + y \equiv 69 \pmod{97} \quad \rightarrow \quad z \equiv 69 \pmod{97}$$

Now three equations can be solved using the Chinese remainder theorem to find z. One of the acceptable answers is $z = 457$.

QUADRATIC CONGRUENCE

Quadratic Congruence is a congruence of the equation of the form $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$.
We limit our discussion to quadratic equations in which
$a_2 = 1$ and $a_1 = 0$, that is equation of the form.

$$\mathbf{x^2 \equiv a \pmod{n}}$$

There are two ways:

1. Quadratic Congruence Modulo a Prime
2. Quadratic Congruence Modulo a Composite

Quadratic Congruence Modulo a Prime

In this, we consider the modulus is a prime number. That is the form. **$x^2 \equiv a \pmod{p}$**

Where p is a prime and 'a' is an integer.

Example 1: Solve the **$x^2 \equiv 3 \pmod{11}$**

Solution: 3 congruent to modulo 11 are 3,14,25 (25 is 5x5 or (-5)x(-5))

The given equation has two solutions:

$x^2 \equiv 25 \pmod{11}$

$x \equiv 5 \pmod{11}$ and $x \equiv -5 \pmod{11}$,

But $-5 \equiv 6 \pmod{11}$

So, the solutions are 5 and 6

Check the result: substitute x=5

$5^2 \equiv 25 = 3 \pmod{11}$

substitute x=6

$6^2 \equiv 36 = 3 \pmod{11}$

Example 2: Solve the **$y^2 \equiv 10 \pmod{13}$**

Solution: The number 10 congruent to 13 are 10,23,36 (36 is 6x6 or (-6)x(-6))

The given equation has two solutions:

$x \equiv$ **$6 \pmod{13}$ and $x \equiv -6 \pmod{13}$,**

But $-6 \equiv 7 \pmod{13}$

So, the solutions are 6 and 7

Check the result: substitute x=6

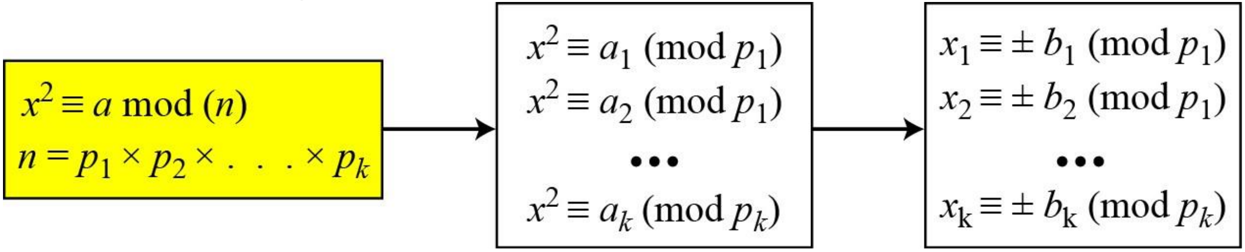
$6^2 \equiv 36 \equiv 10 \pmod{13}$

substitute x=7

$7 \equiv 49 \equiv 10 \pmod{13}$

Quadratic Congruence Modulo a Composite

Quadratic Congruence Modulo a Composite can be solved by set of Quadratic Congruence Modulo a Prime. Decomposition of congruence modulo a composite:



$$x^2 \equiv a \bmod (n) \qquad n = p_1 \times p_2 \times \ . \ . \ . \times p_k$$

$$x^2 \equiv a_1 \pmod{p_1} \qquad x^2 \equiv a_2 \pmod{p_1} \qquad \bullet\bullet\bullet \qquad x^2 \equiv a_k \pmod{p_k}$$

$$x_1 \equiv \pm b_1 \pmod{p_1} \qquad x_2 \equiv \pm b_2 \pmod{p_1} \qquad \bullet\bullet\bullet \qquad x_k \equiv \pm b_k \pmod{p_k}$$

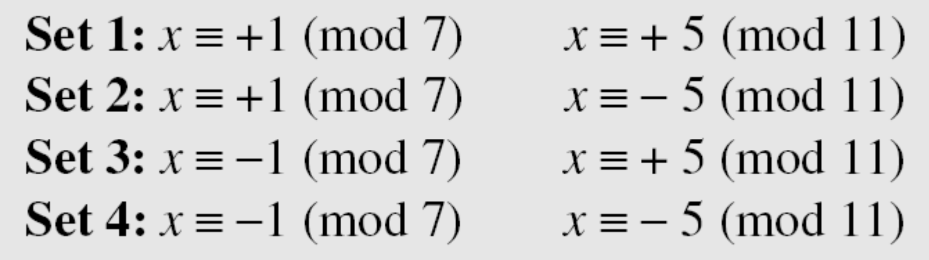
Example: Assume that $x^2 \equiv 36 \pmod{77}$.

We know that $77 = 7 \times 11$. We can write

$$x^2 \equiv 36 \pmod{7} \equiv 1 \pmod{7} \quad \text{and} \quad x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$$

The answers are $x \equiv +1 \pmod{7}$, $x \equiv -1 \pmod{7}$,

$x \equiv +5 \pmod{11}$, and $x \equiv -5 \pmod{11}$. Now we can make four sets of equations out of these:



Set 1: $x \equiv +1 \pmod 7$ $\quad x \equiv +5 \pmod{11}$

Set 2: $x \equiv +1 \pmod 7$ $\quad x \equiv -5 \pmod{11}$

Set 3: $x \equiv -1 \pmod 7$ $\quad x \equiv +5 \pmod{11}$

Set 4: $x \equiv -1 \pmod 7$ $\quad x \equiv -5 \pmod{11}$

The answers are x = ± 6 and ± 27.

ASYMMETRIC KEY /PUBLIC KEY CRYPTOGRAPHY

Asymmetric key cryptosystems / public-key cryptosystems use a pair of keys: public key (encryption key) and private key (decryption key).

Public Key Cryptography ?

- Public key cryptography also called as **asymmetric cryptography**.
- It was invented by whitfield **Diffie** and Martin **Hellman** in 1976. Sometimes this cryptography also called as **Diffie-Helman Encryption**.
- Public key algorithms are based on mathematical problems which admit no efficient solution that are inherent in certain integer factorization, discrete logarithm and Elliptic curve relations.

Public key Cryptosystem Principles:

- The concept of public key cryptography is invented for two most difficult problems of Symmetric key 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.

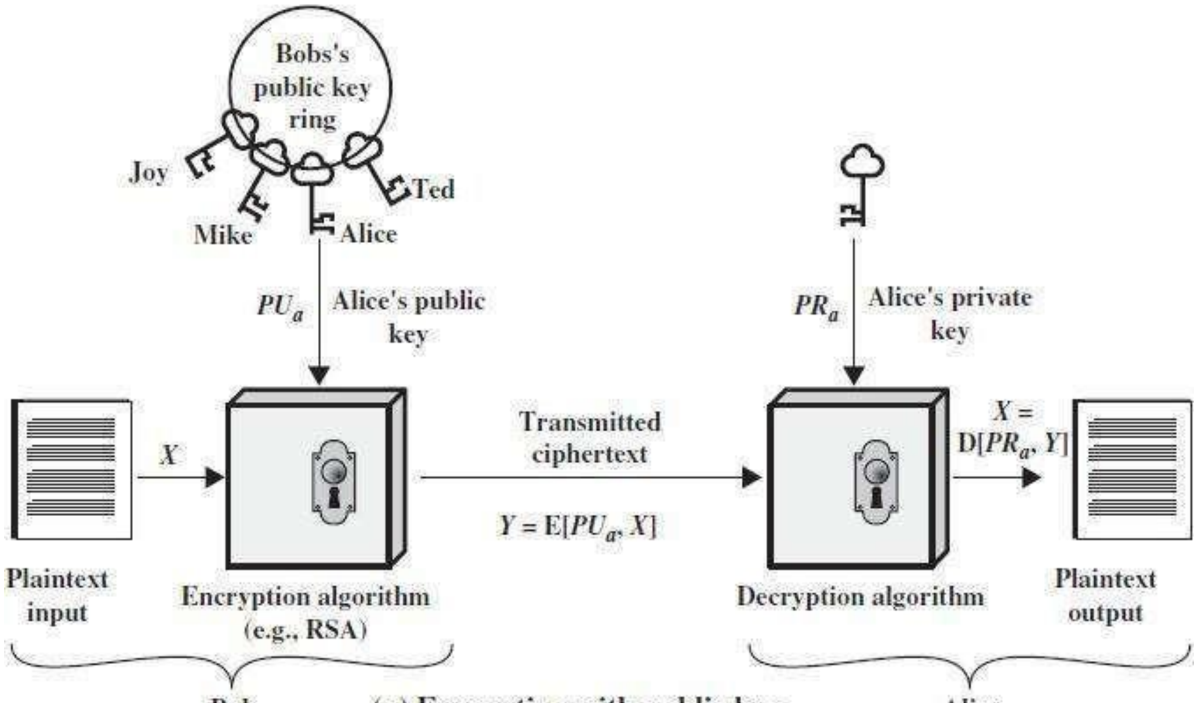
- This public key cryptosystem uses two keys as pair for encryption of plain text and Decryption of cipher text.
- These two keys are names as “**Public key**” and “**Private key**”. The private key is kept secret where as public key is distributed widely.
- A message or text data which is encrypted 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
4	**Private key**	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
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)



(a) Encryption with public key

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 the above Figure 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.

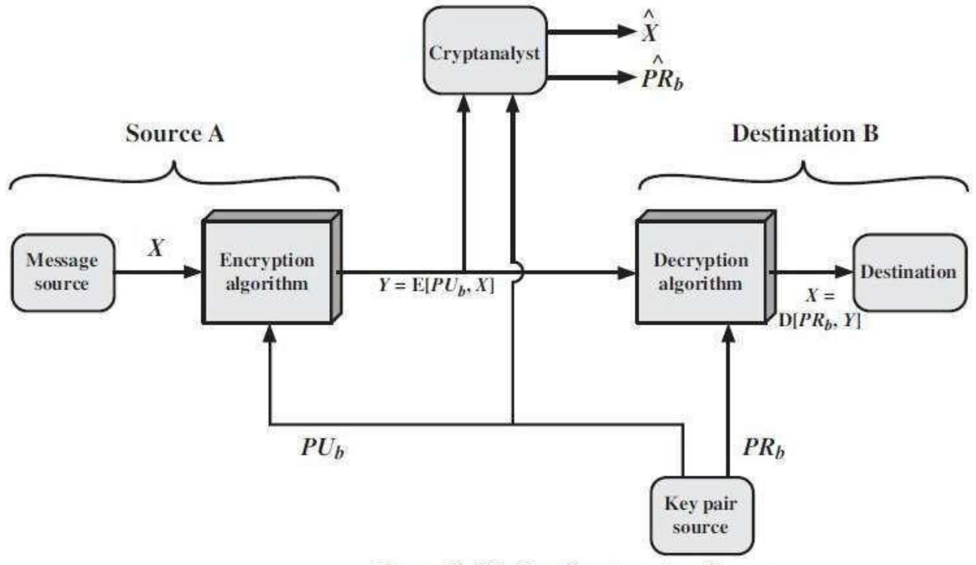


Figure Public-Key Cryptosystem: Secrecy

There is some source A that produces a message in plaintext **$X = [X_1, X_2, . . . ,X_M]$.**

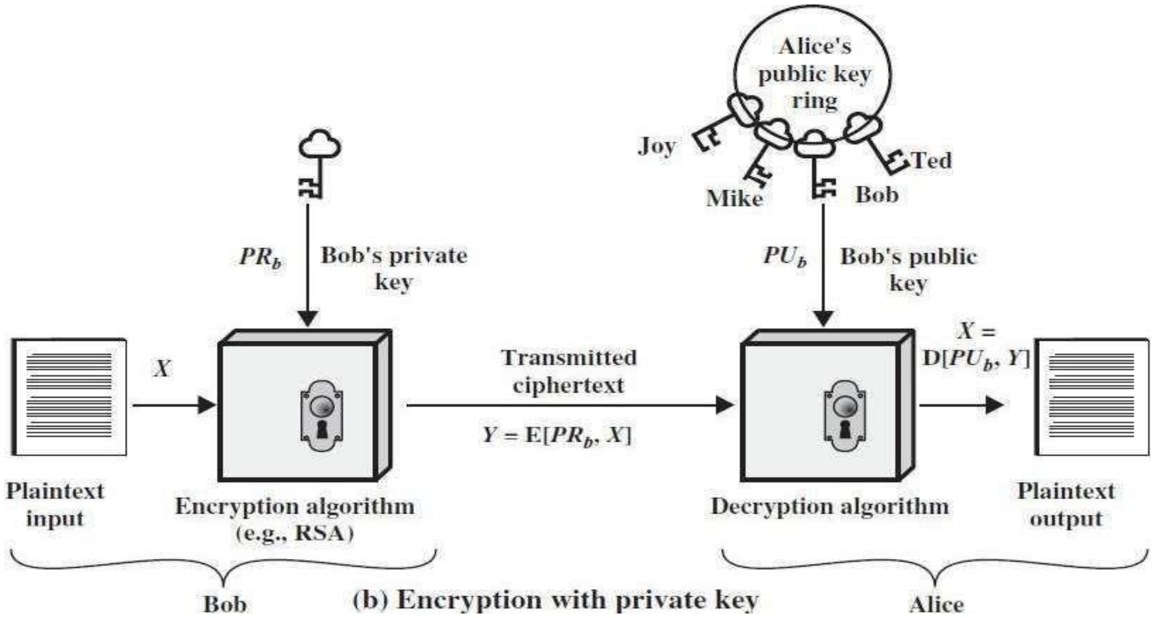
The ***M*** elements of ***X*** are letters in some finite alphabet. The message is intended for destination **B**. B generates a related pair of keys: a public key, ***PU_b*,** and a private key, ***PR_b***.
PR_b is known only to B, whereas ***PU_b*** is publicly available and therefore accessible by A.
With the message *X* and the encryption key ***PU_b*** as input, A forms the ciphertext $Y = [Y_1, Y_2, . . . , YN]$:

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

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

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

Public key cryptography for proving Authentication:



(b) Encryption with private key

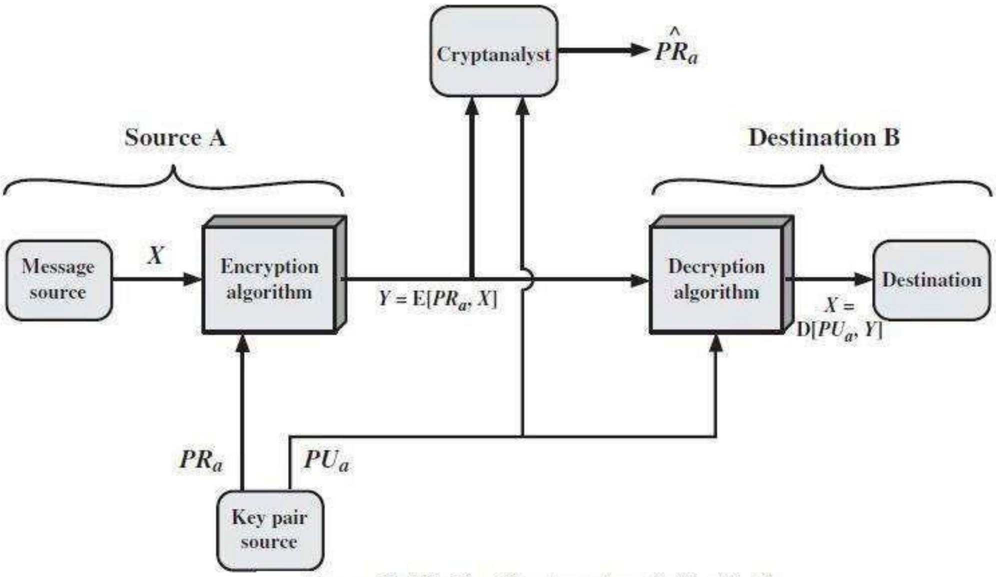


Figure Public-Key Cryptosystem: Authentication

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.

Public key cryptography for both authentication and confidentiality (Secrecy)

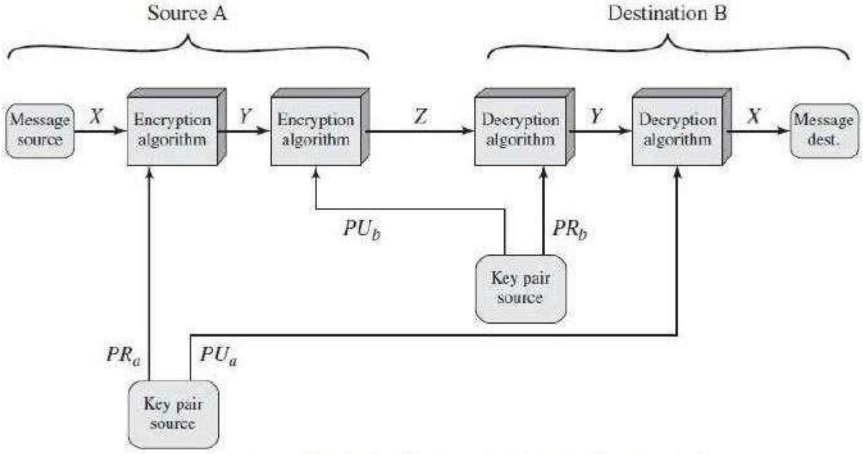


Figure Public-Key Cryptosystem: Authentication and Secrecy

It is, however, possible to provide both the authentication function and confidentiality by a double use of

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

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

the public-key scheme (above figure):
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.

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. 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. However, there is a tradeoff to be considered. Public- key systems depend on the use of some sort of invertible mathematical function. The complexity of calculating these functions may not scale linearly with the number of bits in the key but grow more rapidly than that. 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 the most common public key algorithm.
- This RSA name is get from its inventors first letter (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 **log₂(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

$$\boxed{C = M^e \bmod n}$$

$$\boxed{M = C^d \bmod n}$$

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

Both sender and receiver must know the value of n.

The sender knows the value of **e** & only the receiver 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}

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., e with gcd (Ø(n), e)=1 where 1<e< Ø(n).
Step 5→ Calculate "d" (private key) by using following condition.
d< Ø(n).
Step 6→ Perform encryption by using $\boxed{ed \equiv 1 \bmod Ø(n)}$
Step 7→ performDecryption by using $\boxed{M = C^d \bmod n}$

Example:

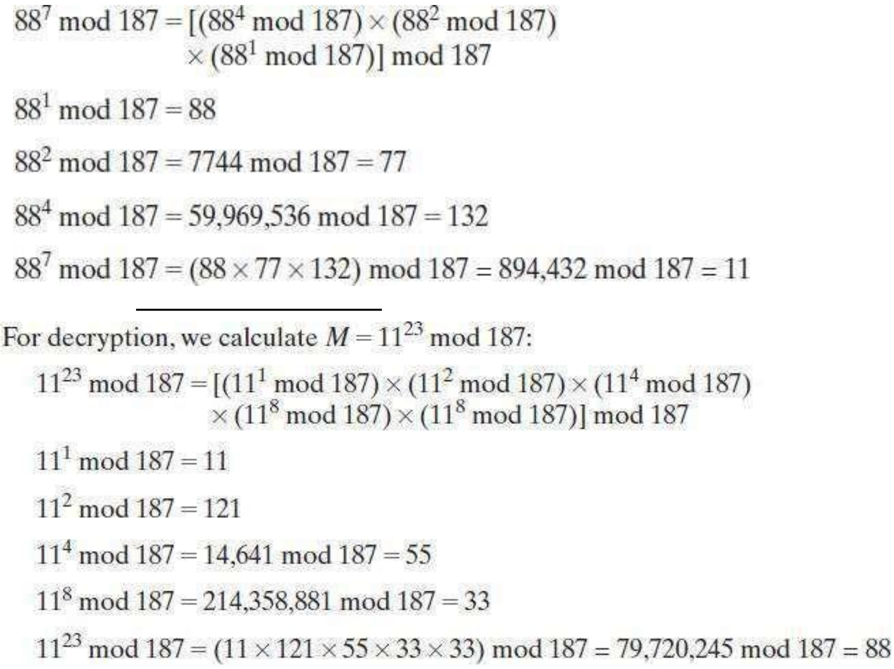
1. Select two prime numbers, *p* = 17 and *q* = 11.
2. Calculate *n* = *pq* = 17 × 11 = 187.
3. Calculate Ø(*n*) = (*p* - 1)(*q* - 1) = 16 × 10 = 160.
4. Select *e* such that *e* is relatively prime to Ø(*n*) = 160 and less than Ø (*n*); we choose *e* = 7.
5. Determine *d* such that ***de* ≡1 (mod 160)** and *d* < 160.The correct value is *d* = 23, because 23 * 7 = 161

= (1 × 160) + 1;

d can be calculated using the extended Euclid‟s algorithm

6. 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 $C = 88^7$ mod 187. Exploiting the properties of modular arithmetic, we can do this as follows.



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

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

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

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

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

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

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

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

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

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

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

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

The Security of RSA

Four possible approaches to attacking the RSA algorithm are

- **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.

Trapdoor one-way function

- A trapdoor function is a function that is easy to perform one way, but has a secret that is required to perform the inverse calculation efficiently.
- That is, if f is a trapdoor function, then y=f(x) is easy to compute, but x=f−1(y) is hard to compute without some special knowledge k. Given k, then it is easy to computey=f−1(x,k).
- The analogy to a "trapdoor" is something like this: It's easy to fall through a trapdoor, but it's very hard to climb back out and get to where you started unless you have a ladder.
- An example of such trapdoor one-way functions may be finding the prime factors of large numbers. Nowadays, this task is practically infeasible.
- On the other hand, knowing one of the factors, it is easy to compute the other ones.

For example: RSA is a one-way trapdoor function

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 from 1 to P-1 that is if **'a'** is a primitive root of the prime number P, then the numbers 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

$b \equiv a^i \bmod P \qquad 0 \le i \le (P-1)$

The exponent i → is refer as discrete logarithm or index of b for the base a, mod P.
The value denoted as **ind** $_{a,p}$**(b)**

Algorithm for Diffie-Hellman Key Exchange:

Step 1→ Select global public 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

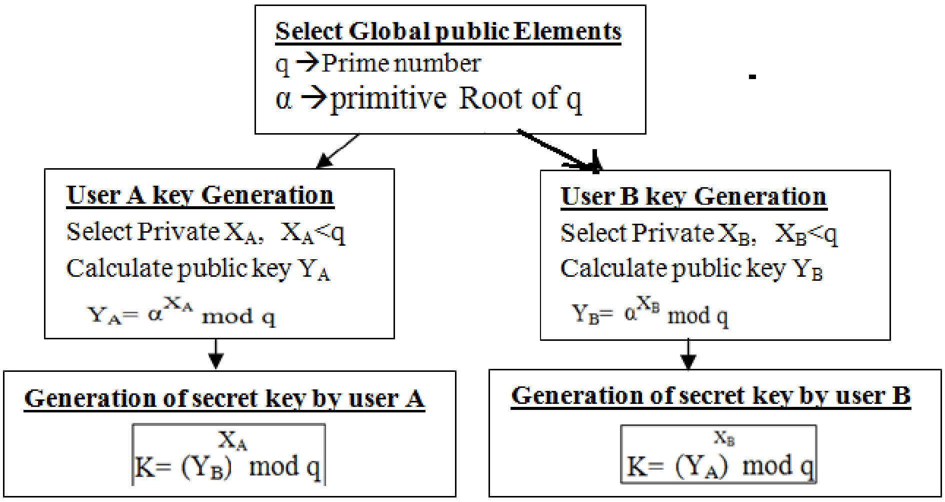
a) User A select a random integer X_A<q and computes $Y_A = \alpha^{X_A} \bmod q$

b) User B independently select a random integer X_B <q and computes $Y_B = \alpha^{X_B} \bmod q$

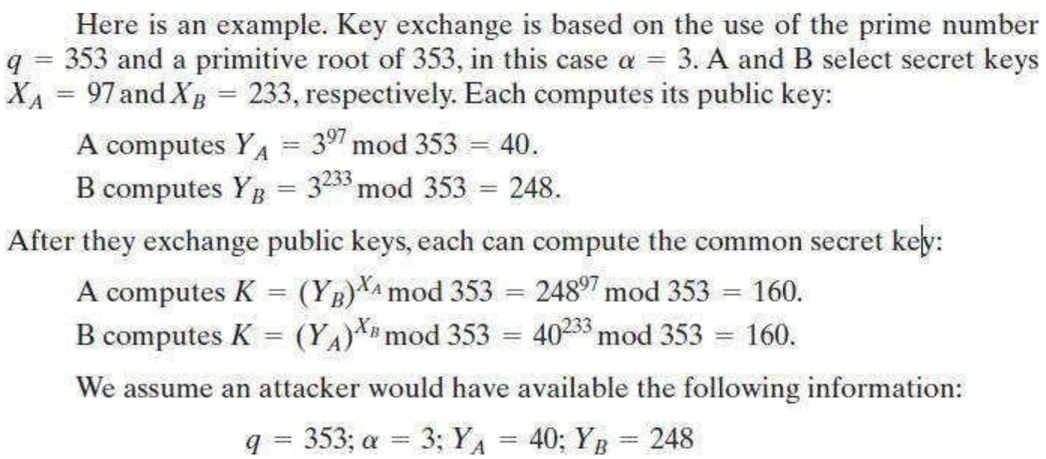
c) Each side keeps the X value private and Makes the Y value available publicly to the outer side.

Step 3→ User A Computes the key as $\boxed{K = (Y_B)^{X_A} \bmod q}$
User B Computes the key as $\boxed{K = (Y_A)^{X_B} \bmod q}$

Step 4→ two calculation produce identical results
The result is that the two sides have exchanged a secret key.


Select Global public Elements
q →Prime number
α →primitive Root of q
User A key Generation
Select Private X_A, X_A<q
Calculate public key Y_A
$Y_A = \alpha^{X_A} \bmod q$
User B key Generation
Select Private X_B, X_B<q
Calculate public key Y_B
$Y_B = \alpha^{X_B} \bmod q$
Generation of secret key by user A
$K = (Y_B)^{X_A} \bmod q$
Generation of secret key by user B
$K = (Y_A)^{X_B} \bmod q$

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

We assume an attacker would have available the following information:

$$q = 353;\ \alpha = 3;\ Y_A = 40;\ Y_B = 248$$

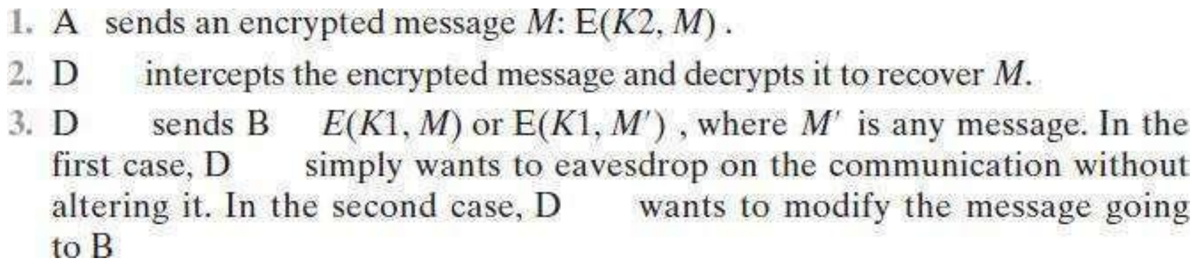
MAN-in the Middle Attack (MITM)

Definition: A man in the middle attack is a form of eavesdropping where communication between two users is monitored and modified by an unauthorized party.
Generally the attacker actively eavesdrops by intercepting (stoping) a public key message exchange.

The Diffie- Hellman key exchange is insecure against a "Man in the middle attack".
Suppose user A & B wish to exchange keys, and D is the adversary (opponent). The attack proceeds as follows.

1. D prepares for the attack by generating two random private keys X_{D1} & X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. A transmits Y_A to B
3. D intercepts Y_A and transmits Y_{D1} to B. and D also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. B receives Y_{D1} & calculate $K1 = (Y_{D1})^{X_B} \bmod q$.
5. B transmits Y_B to A
6. Dintercepts Y_B and transmits Y_{D2} to „A" and „D" calculate K1 $K1 = (Y_B)^{X_{D1}} \bmod q$
7. A receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key *K*1 and Alice and Darth share secret key *K*2. All future communication between Bob and Alice is compromised in the following way.



1. A sends an encrypted message *M*: E(*K*2, *M*) .
2. D intercepts the encrypted message and decrypts it to recover *M*.
3. D sends B *E*(*K*1, *M*) or E(*K*1, *M*′) , where *M*′ is any message. In the first case, D simply wants to eavesdrop on the communication without altering it. In the second case, D wants to modify the message going to B

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.