WHY ENCRYPTION?

Objectives of cryptography:

There are four main objectives of cryptography:-

a). Confidentiality: It guarantees that the sensitive information can only be accessed by those users/entities authorized to unveil it.

b). Data integrity: It is a service which addresses the unauthorized alteration of data. This property refers to data that has not been changed, destroyed, or lost in a malicious or accidental manner.

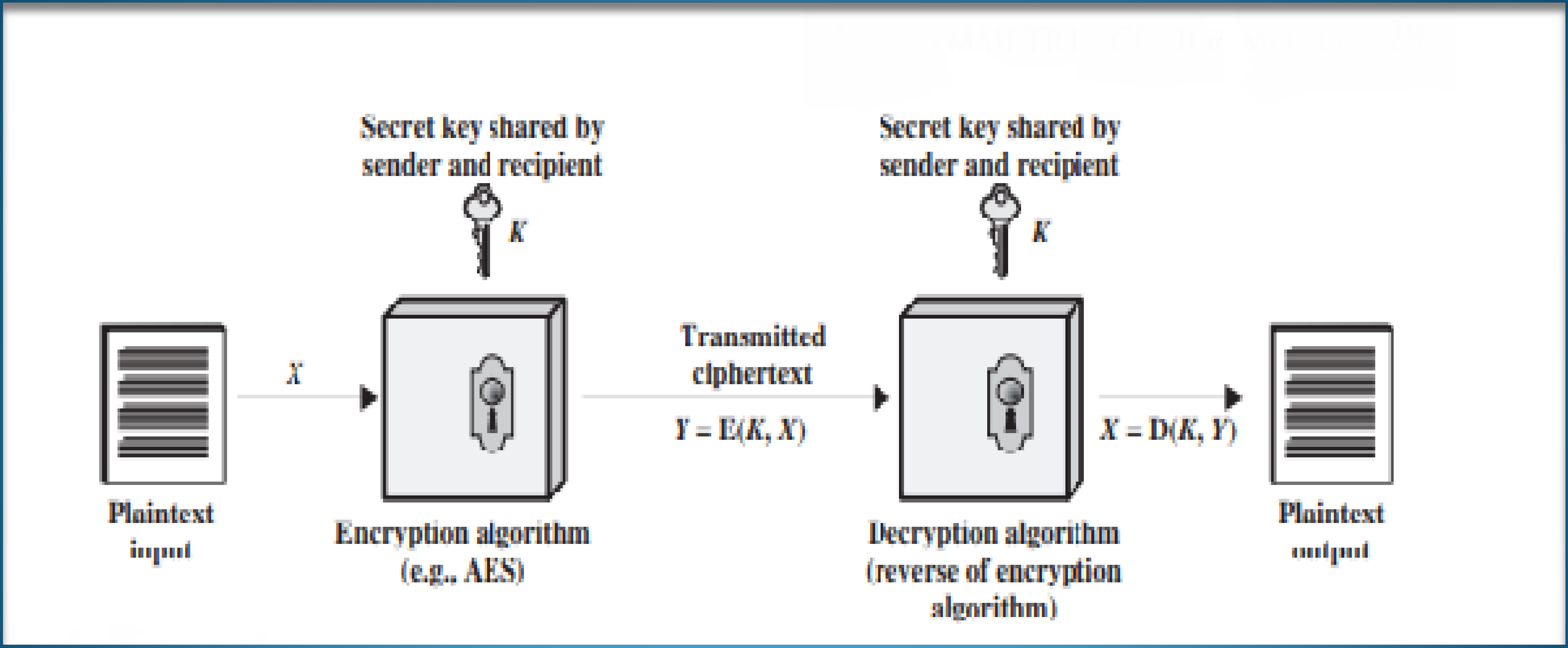
c). Authentication: It is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other.

d). Non-repudiation: It is a service which prevents an entity from denying previous commitments or actions.

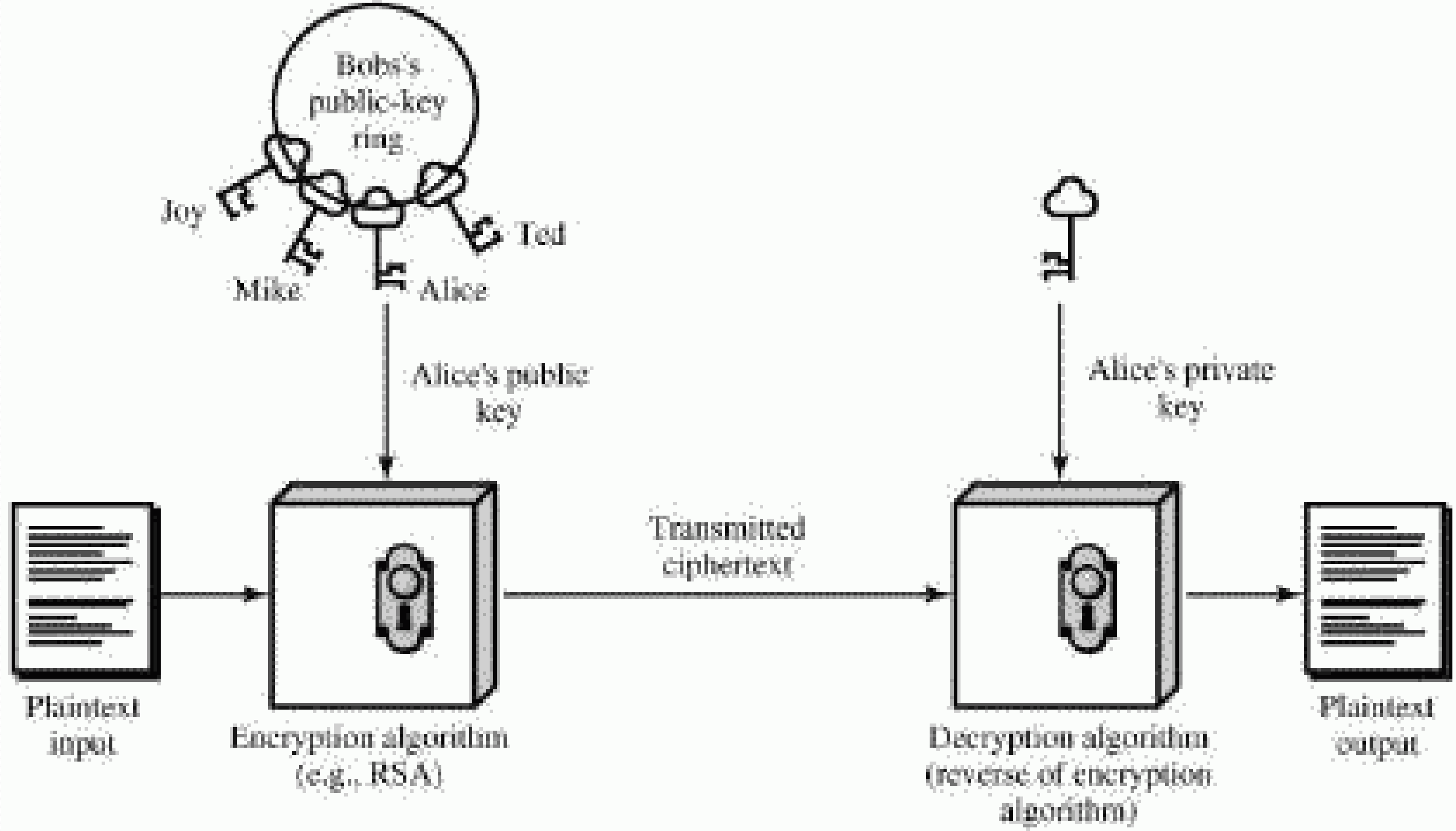
TYPES

- Cryptography involves two main approaches:
- Symmetric-key cryptography.
- Asymmetric-key cryptography.
- **Symmetric-key cryptography: Same secret key is used for**
- both encryption and decryption.
- **Asymmetric-key cryptography: Two different keys are used**
- i.e. one for encryption and other for decryption.

SIMPLIFIED MODEL OF SYMMETRIC CRYPTO SYSTEM



SIMPLIFIED MODEL OF ASYMMETRIC CRYPTO SYSTEM



Comparison of Secret Key and Public Key

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of Key	1	2
Protection of Key	Must be kept secret	One key must be kept secret & other can be freely exposed
Best Uses	secrecy and integrity of data	Key exchange, authentication
Key Distribution	Problematic	Safer
Speed	Fast	Slow; typically, 10,000 times slower than secret key

RSA

Key Generation by Alice	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

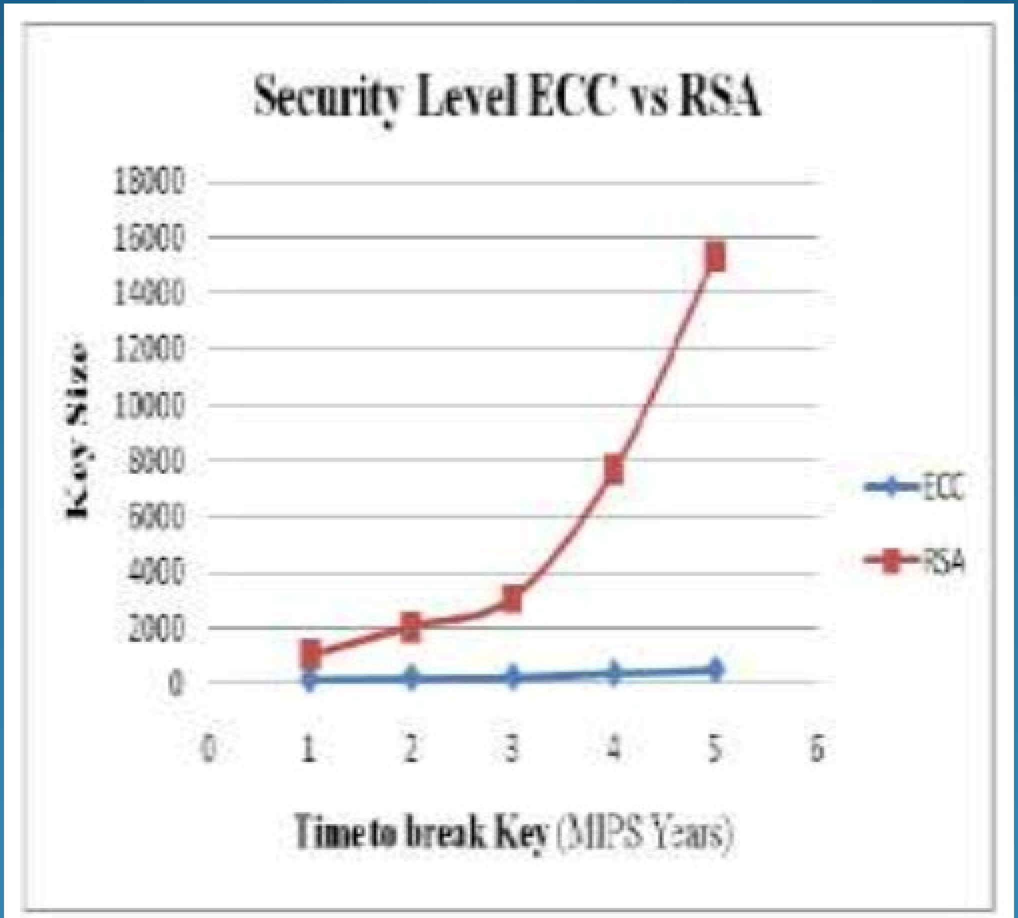
Encryption by Bob with Alice's Public Key	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

The main interest of the elliptic curve cryptosystems is to decrease the required key-size to achieve appropriate security. Thus, we have an equivalent security level among RSA algorithm using 1024-bit key and an elliptic curve cryptosystem using more or less a 160-bit key. The following comparison table given by NIST in [SP800-57] perfectly illustrates the key size benefits of using elliptic curves based cryptography:

Symmetric Key Algorithms	Diffie-Hellman, Digital Signature Algorithm	RSA (size of n in bits)	ECC (modulus size in bits)
80	$L = 1024$ $N = 160$	1024	160–223
112	$L = 2048$ $N = 224$	2048	224–255
128	$L = 3072$ $N = 256$	3072	256–383
192	$L = 7680$ $N = 384$	7680	384–511
256	$L = 15{,}360$ $N = 512$	15,360	512+

Note: L = size of public key, N = size of private key


Security Level ECC vs RSA
Key Size
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6
Time to break Key (MIPS Years)
ECC
RSA

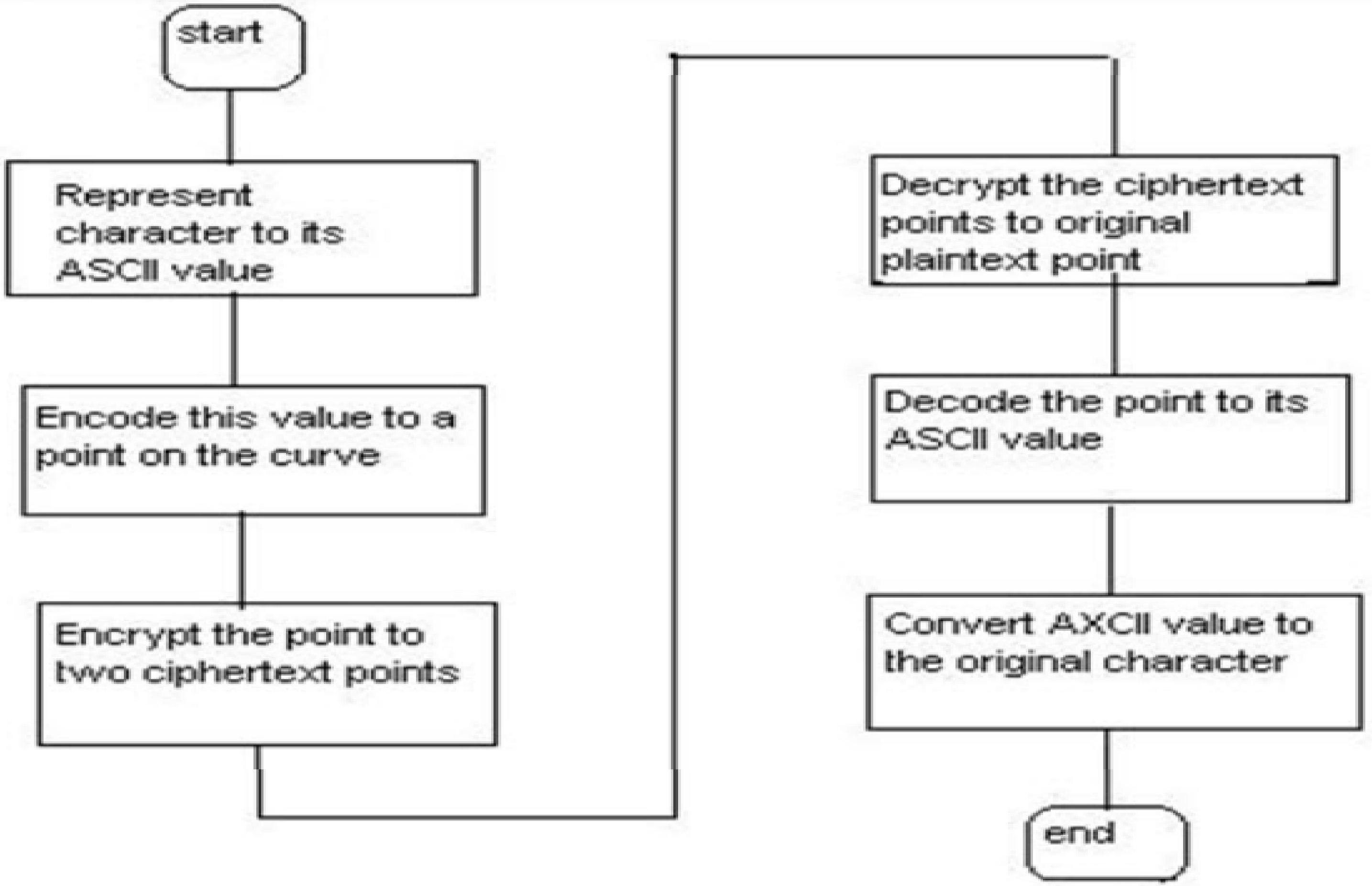
ECC

RS
A

ECC-What is it?

- Elliptic curves are not ellipses. Elliptic curves are described by cubic equations similar to those used for calculating the circumference of an ellipse
- Elliptic curve cryptography makes use of elliptic curves, in which the variables and coefficients are all restricted to elements of a finite field
- An elliptic curve is defined by an equation in two variables with coefficients.

HOW IT RUNS?



start
Represent character to its ASCII value
Encode this value to a point on the curve
Encrypt the point to two ciphertext points
Decrypt the ciphertext points to original plaintext point
Decode the point to its ASCII value
Convert AXCII value to the original character
end

y^2mod p = (x^3+ ax + b) mod p

Lists of points (other than O) that are part of E_{23} (1, 1). plots the points of E_{23}(1, 1); note that the points, with one exception, are symmetric about y = 11.5.

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

Koblitz's Method for Encoding Plaintext

Step1: Pick an elliptic curve Ep(a,b).

Step 2: Let us say that E has N points on it.

Step 3: Let us say that our alphabet consists of the digits 0,1,2,3,4,5,6,7,8,9 and the letters A,B,C,. . . , X,Y,Z coded as 10,11,. . . , 35.

Step 4: This converts our message into a series of numbers between 0 & 35.

Step 5: Now choose an auxiliary base parameter, for example k = 20. (both parties should agree upon this)

Step 6: For each number mk (say), take x=mk + 1 and try to solve for y.

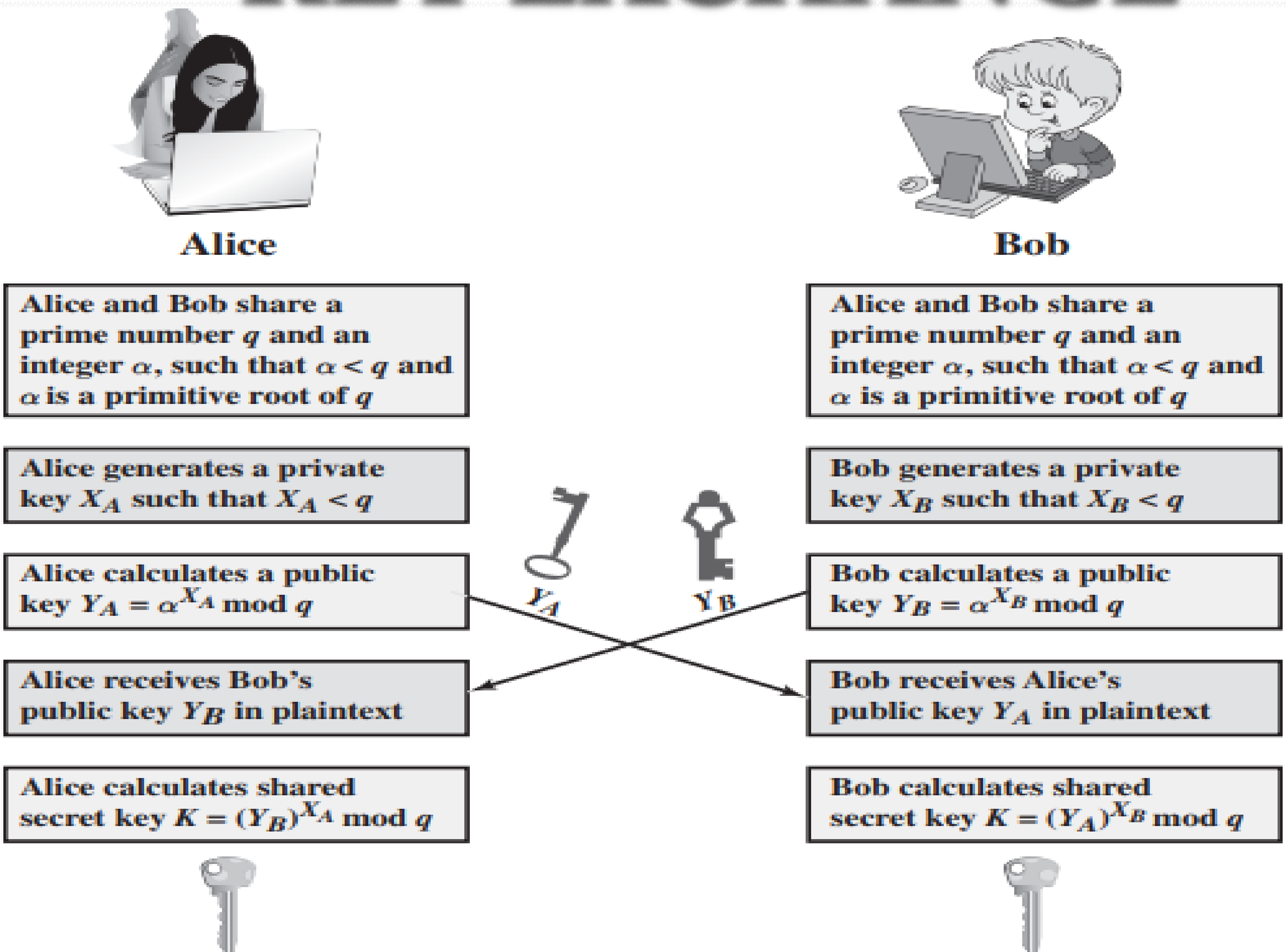
Step 7: If you can't do it, then try x = mk +2 and then x = mk +3 until you can solve for y.

Step 8: In practice, you will find such a 'y' before you hit x = mk + k - 1. Then take the point (x,y). This now converts the number m into a point on the elliptic curve. In this way, the entire message becomes a sequence of points.

HOW IS KEY GENERATED?

- Both the entities in the cryptosystem agree upon a,b,p,G,n which are called �Domain Parameters�of G is called generator point and n is the order of G.
- Now A generates a random number nA< n as his private Key and calculates his public key
- Set P_A= G+G+G�.....nA times.
- B generates a random number nB< n as his private Key and calculates his public key.
- set P_B= G+G+G�....nB times.

KEY EXCHANGE



ENCRYPTION

A sends C = 2 ciphertext points those are { kG, P_m + kP_b) }.

Where G - generator Point

P_m - plaintext point on the curve

k - a random number chosen by A

P_b - public key of B

DECRYPTION

$$P_m + kP_B - n_B(kG) = P_m + k(n_B)G - n_B\,(kG) = P_m$$

Where G - generator Point

P_m - plaintext point on the curve

k - a random number chosen by A

P_B - public key of B

AN EXAMPLE

Example: Say the parameters of curve are: p(751),a(-1),b(188),n(727).

1. Say we have to send character ‘b’.
2. ‘B’ is first encoded as number 11.
3. x=mk+1 i.e, 11*20+1=221cannot solve it for a y such that y^2 = x^3 + ax+ b mod p
4. So go for x=mk+2 , x=222 , no y exists. x=mk+3, x=223, no y exists.
5. x=mk+4 so x=224 can solve it for y and y=248.
6. Now the point (224,248) is point is encrypted and decrypted as a message.
7. To decode just compute (x-1)/k i.e, (224-1)/20=223/20 i.e, 11.15.
8. Return 11 as original plaintext(greatest integer less than (x-1)/k ,that is 11.
9. The number 11 is now decoded to character ‘B’.
10. The probability that we fail to find a square (and hence fail to associate m to a point) is about (½)^k .

OUR PROGRESS

- function [X,Y,n] = **PC**(A,B,p)

This function m-file finds and plots all the points that lie in E_p(A,B)
These points are on the curve y^2 = x^3 + AX + B (mod p)

- function [x3,y3,m] = **ECADP**(x1,y1,x2,y2,A,p)

This function m-file performs Elliptic Curve addition over prime curves.
Suppose we are working on the elliptic curve y^2 = x^3 + Ax + B
Define P1 = (x1,y1)
P2 = (x2,y2)
Then P1 + P2 = P3 = (x3,y3) is defined by as below
If one if the variables in infinity then we define P + infinity = P
and the user should type in 'infinity' for both the x and y values.

- function [X2,Y2] = **SUCDOB**(X1,Y1,k,A,p)

This is a function m-file to perform the successive doubling algorithm on prime curves. If P = (X1,Y1) and k is an integer, then this algorithm will find kP = (X2,Y2) where we are operating over the elliptic curve
y^2 = x^3 + Ax + B (mod p), p prime

OUR PROGRESS

- function [flag] = **check**(x,y,A,B,p)

An m-file to check if the point (x,y) lies on the prime curve
y^2 = x^3 + Ax + B (mod p)

- function [I] = **inve**(N,p)

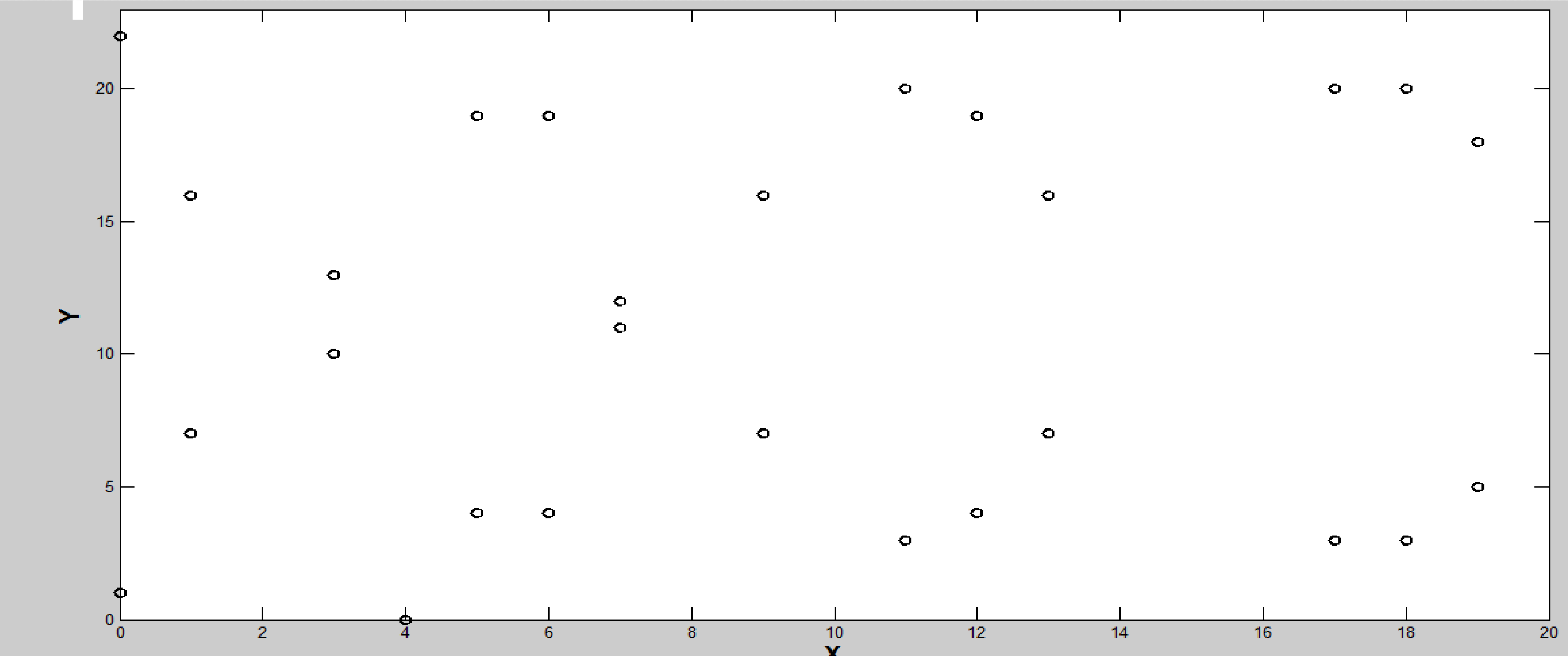
This m-file finds the inverse of an element, N, in the group Z_p
for use with prime curves.

- function [x10,y,c] = **ECC**(x,y,a,b,M,x9,n)
- function [c1,c2,X6,Y6]=**encrypt**(X2,Y2,X,Y,a,p,n)
- function [X14,Y14,c,str_back_to_char]=**decrypt**(x1,y1,X,Y,n,a,b,p,z,c)
- **Main function**
- M = input('\nEnter the message: ','s'); **% ASCII TO MESSAGE CONVERSION**

Let us look more closely

Output of PC.m

The set of points for a=b=1 and p=23



Output of PC.m

```
Command Window
New to MATLAB? Wa
>> PC(1,1,23)

ans =

     4
     0
     0
    19
    19
     1
     1
     9
     9
    13
    13
     7
     7
     3
     3
    11
    11
    17
    17
    18
    18
     5
     5
     6
     6
    12
    12
```

Output of ECADP.m

Command Window

New to MATLAB? Watch this Video, se

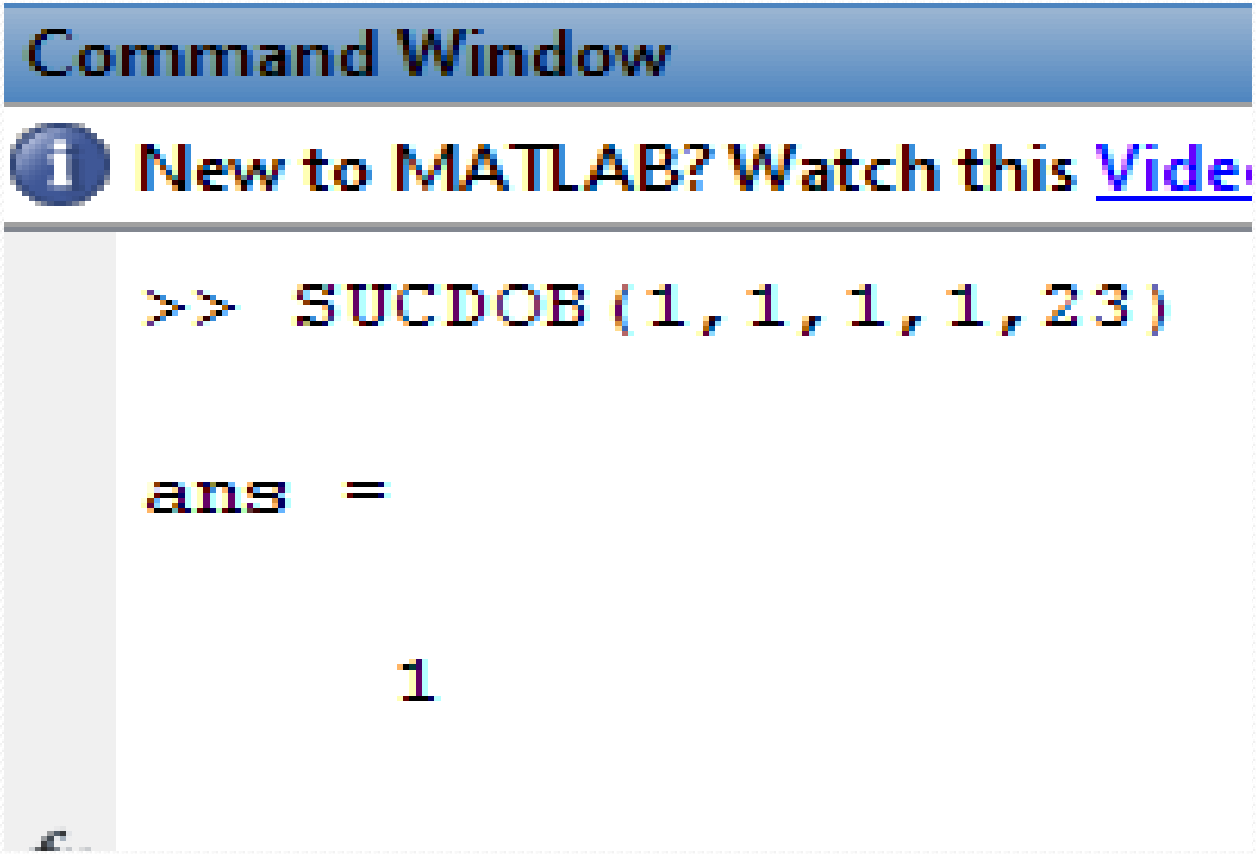
```
>> ECADP(1,1,2,3,1,23)

ans =

     1

>> 
```

Output of SUCDOB.m



Output of check.m

Command Window

New to MATLAB? Watch this Video, see Demos, or read Ge

```
>> check(1,1,1,1,23)
This point does not lie on the curve


ans =


'NO


>>
```

Output of inve.m

Command Window

New to MATLAB? Watch this Video, see

```
>> inve(23,23)

ans =

   Empty matrix: 0-by-1

>>
```

Output of encrypt.m

Command Window

New to MATLAB? Watch this Video, see De

```
>> encrypt(1,1,1,1,1,23,23)
ciper text 1
     2


     3

P3 is infinity
ciper text 2
     1


     1



ans =

     2
```

Output of decrypt.m

Command Window

New to MATLAB? Watch this Video, see Demos, or

```
>> decrypt(1,1,1,1,23,1,1,23,2,3)
decrypted
         1


         1



str_back_to_char =

□

□

ans =

         1
```

ASCII to Message

Command Window

New to MATLAB? Watch this Video, see Demos, or read Ge

```
Enter the message: soham

str_ascii =

   115   111   104    97   109


str_back_to_char =

soham


str_16bit =

     115     111     104      97     109


str_back_to_char =

soham
```

ECC implementation in Matlab

```
p=23;
a=1;b=1;
[x,y,n]=PC(1,1,p);
x1=x(7);
y1=y(7);
z=randi([1,n-1]);
[X2,Y2]=SUCDOB(x1,y1,z,1,p);
M = input('\nEnter the message: ','s');
x9=length(M);
[X,Y,c]=ECC(x1,y1,a,b,M,x9,n);
na=randi([1,n-1]);

[c1,c2,X6,Y6]=encrypt(x1,y1,X,Y,a,p,n);
[X14,Y14,c5,str_back_to_char]=decrypt(x1,y1,X,Y,n,a,b,p,z,c);
```

Final Output

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
Enter the message: SOHAM CHAYAN BISHWADIP
point does not lie on curve
ASCII Code of the entered Message:
    83    79    72    65    77    32    67    72    65    89    65    78    32    66    73    83    72    87    65    68    73    80

encrypted message:
   602

    16

ciper text 1
    18

    20

ciper text 2
     1

     3

decrypted
   602

    16


str_back_to_char =

SOHAM CHAYAN BISHWADIP
```

Presented By- SOHAM MONDAL
BISHWADIP PAUL
CHAYAN DUTTA

Dept of ETCE(UG-III)