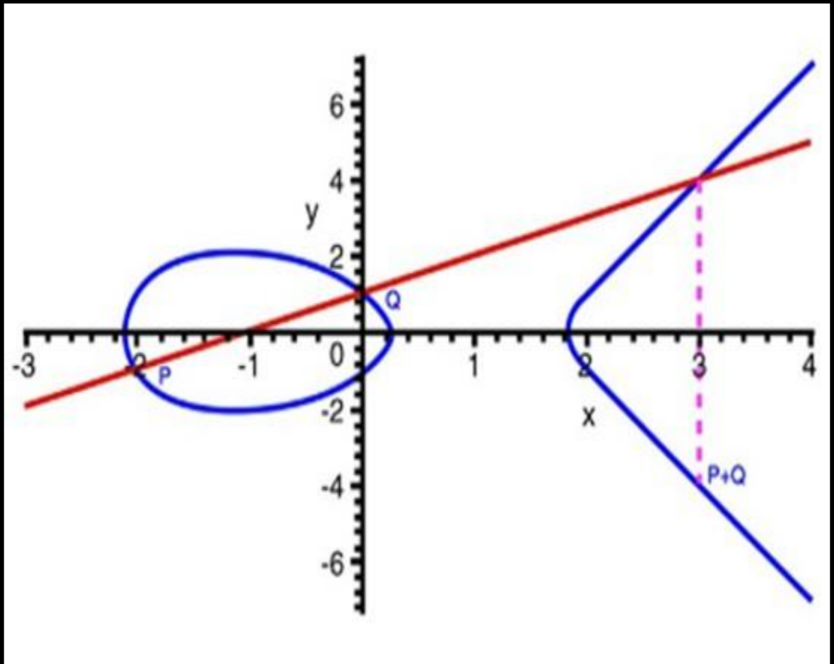
ELLIPTIC CALCULATOR



By:
Name: Indrajit Mondal
Roll : 000911001013
Name : Jayati De
Roll : 000911001046
Dept of Information Technology,
Jadavpur University

References

- Lecture Notes & other materials supplied to us by our Mentor Prof. Utpal Kr. Ray.
- Cryptography & Network Security by B A Forouzan , D Mukhopadhyay
- http://www.ccs.neu.edu/home/riccardo/courses/cs6750-fa09/talks/Ellis-elliptic-curve-crypto.pdf
- http://www.certicom.com/index.php?action=ecc_tutorial,home
- http://cse.unl.edu/~xkzou/CSE477/
- Cryptographic information from Wikipedia, URL http://en.wikipedia.org/wiki/Cryptography

CRYPTOGRAPHY

- Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription. Some experts argue that cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about *any* network, particularly the Internet.

TYPES OF CRYPTOGRAPHIC ALGORITHMS

There are several ways of classifying cryptographic algorithms. For purposes of this paper, they will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms that will be discussed are (Figure on next slide):

- Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption
- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
- Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information

Three types of Cryptography: Secret-key, Public key & Hash function

A) Secret key (symmetric) cryptography. SKC uses a single key for both encryption and decryption.

B) Public key (asymmetric) cryptography. PKC uses two keys, one for encryption and the other for decryption.

C) Hash function (one-way cryptography). Hash functions have no key since the plaintext is not recoverable from the ciphertext.

Elliptic curve cryptography (ECC) :

- Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms that is much more difficult to challenge at equivalent key lengths.

Advantages:

- greater flexibility in choosing cryptographic system.
- We can use keys of smaller length as compared to other public Key Algorithms like RSA,El-Gamal etc.The minimum key size for ECC should be 132 bits vs. 952 bits for RSA.

Comparison of Key Sizes of Various Public Key Crytography Algorithms :

Elliptic-Curve Digital Signature Algorithm (ECDSA)

NIST Guidelines for Public Key Sizes for AES

ECC key size (bits)	RSA key size (bits)	Key size ratio	AES key size (bits)
163	1,024	1:6	
256	3,072	1:12	128
384	7,680	1:20	192
512	15,360	1:30	256

Supplied by NIST to ANSI X9F1

Table 1

Disadvantages:

- Hyperelliptic cryptosystems offer even smaller key sizes.
- ECC is mathematically more subtle than RSA or SDL) difficult to explain/justify to the client.
- Prone to Brute Force attack .

Applications Of ECC:

- *For shorter key length, ECC has more advantages: higher speeds, lower power consumption, bandwidth savings & storage efficiencies.*
- *These advantages are particularly beneficial in applications where bandwidths, processing capacity, power availability, or storage are constrained. Such applications include Chip card, Electronic commerce, Web servers,Cellular telephones, Pagers etc.*
- *ECC is mainly used in Key exchange mechanism, Digital signature & certificate.*

Elliptic Curve :

- An ***Elliptic Curve*** is a curve given by an equation

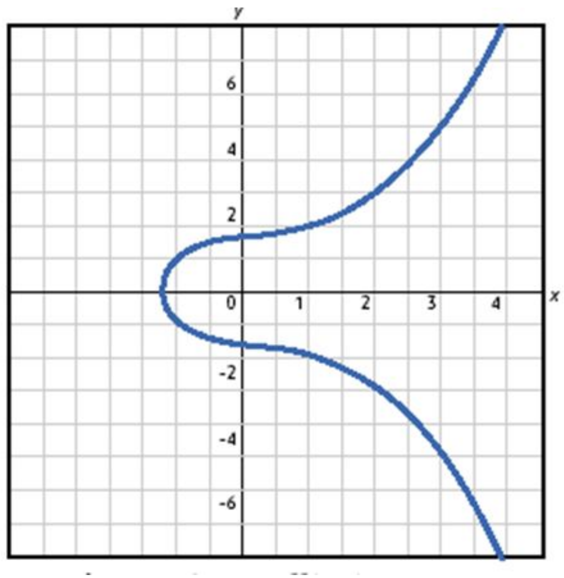
$E : y^2 = f(x)$

Where f(x) is a square-free (no double roots) cubic or a quartic polynomial

After a change of variables it takes a simpler form:

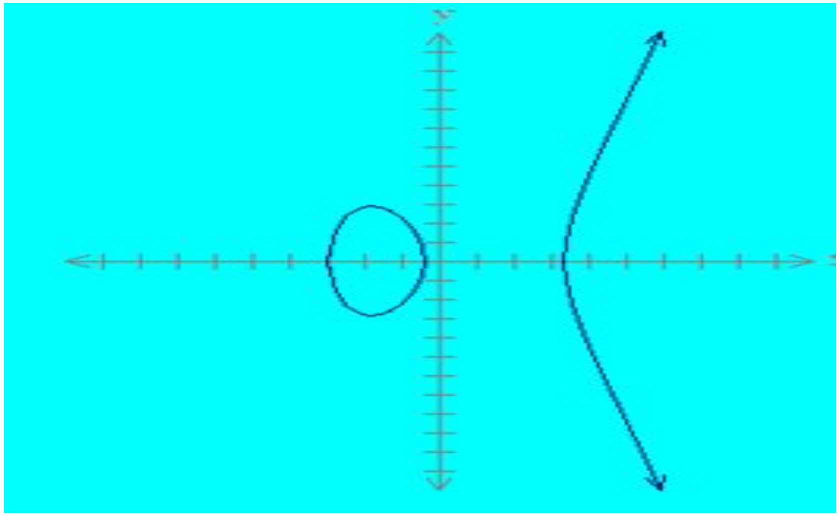
$E : y^2 = x^3 + Ax + B$

So y2 = x3 is not an elliptic curve but y2 = x3-1 is



Elliptic Curve Over Real Field:

- *Elliptic curve equation is:* $y^2 = x^3 + ax + b$ *where a,b are real numbers. An EC over a prime field m is denoted as* $E_m(a,b)$ *.*
- *The curve of this Elliptic curve* **$y^2 = x^3 - 10x - 4$ is :-**



Elliptic Curve Group Over Real Field:

It is an additive group. Its basic operation is addition.

Arithmetic Over Real Field:

- O serves as additive identity:
- O= -O; For any point P on the elliptic curve,
- P+O=P. Where, P!=O.
- The negative of point p:
- If P=(x,y), then –P=(x,-y) So, P + (-P) = P - P = O.

Key Generation :

- Agree on the following (public):
- Curve parameters (a, b)
- The modulus p
- Base point G (on the curve)
- Pick a random integer n as private key
- Calculate public key P = n*G

Encryption:

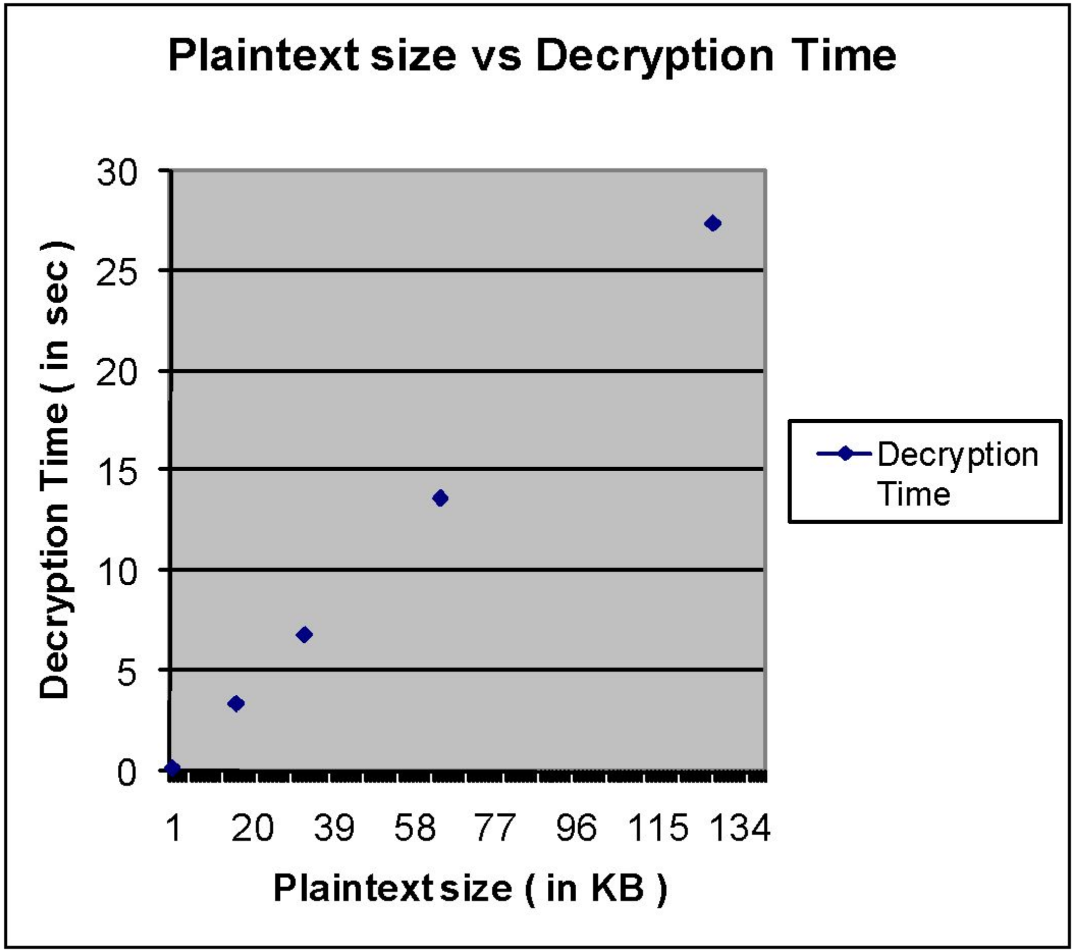
- Alice represents her text or data to send as a point Pm
- Alice sends Bob a pair of points:
- SentPair = {k*G, Pm + k*P}
- k = randomly chosen integer

Decryption:

- Bob decrypts the message using his private key:

$$Pm + k*P - n(k*G) = Pm + k(n*G) - n(k*G) = \mathbf{Pm}$$

Plaintext Size Vs Decryption Time Graph: Here private key n_B = 515



Brute-force Attack:

- Brute force: P = (16,5);
- 2P = (20,20);
- 3P = (14,14);
- 4P = (19,20);
- 5P = (13,10);
- 6P = (7,3);
- 7P = (8,7);
- 8P = (12,17);
- 9P = (4,5)
- So, the value of k is 9.

Discrete Logarithm Problem :

- Consider the equation Q = (k * P),
- where Q, P $\in E_m$ (a, b).
- It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P. This is called discrete logarithm problem for elliptic curves.
- Consider the group E_{23} (9,17). Now, what is the discrete logarithm k of Q = (4,5) to the base P = (16,5).

Brute-force Attack:

- Let we have to calculate (k * G) where k=258,
- we can calculate G + G = 2G, 2G + 2G = 4G, 4G + 4G = 8G, 8G + 8G = 16G, 16G + 16G = 32G, 32G + 32G = 64G, 64G + 64G = 128G, 128G + 128G = 256G, 256G + 2G = 258G.
- So here we need only 9 addition not 257 times repetitive addition.
- But when we are given (k * G) and G to determine k, we have to calculate 2G, 3G, 4G…258G and compare all the values with the value of (k * G) for matching. So it will take a large time. For large value of k, it may be impossible to calculate k within valid period.

ELLIPTIC CALCULATOR

- We have developed an application titled "Elliptic Calculator (Version – 2.0)" whose basic functionality is to compute the various Cryptographic functionalities related to ECC with extra added functionalities to the previous version.
- The application is GUI based and has been written in Java using the Netbeans development environment .
- We chose Java because of its platform independent nature and the easy availability of certain methods and classes .
- Netbeans eased the burden of writing codes for developing the GUI . GUIs were easily created and edited (when required) in the Netbeans IDE.
- We used the Java Desktop Appication for creating the GUIs.
- We have used the BigInteger class to store data inorder to store large and very large values quite easily & accurately.

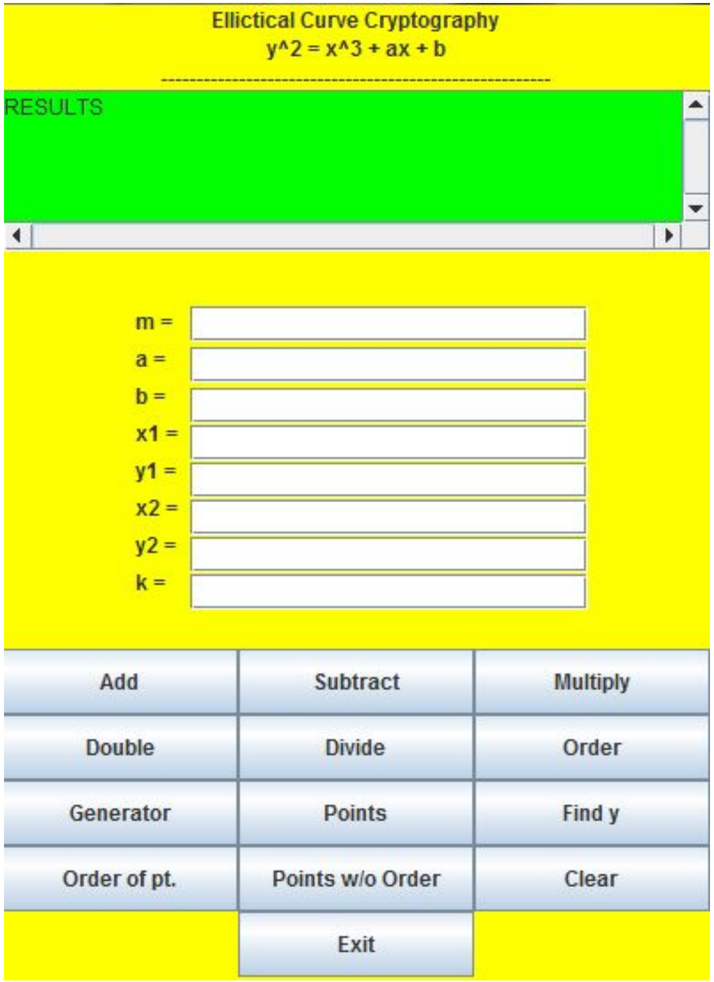
APPLICATIONS OF THIS ELLIPTIC CALCULATOR

- This application can be used for the various ECC related calculations .
- Whenever an ECC based cryptosystem is developed , it is necessary to check its functionalities for any discrepencies . This application can be used effectively to serve this purpose of cross checking the results obtained to see whether there is any fault with the cryptosystem or not .

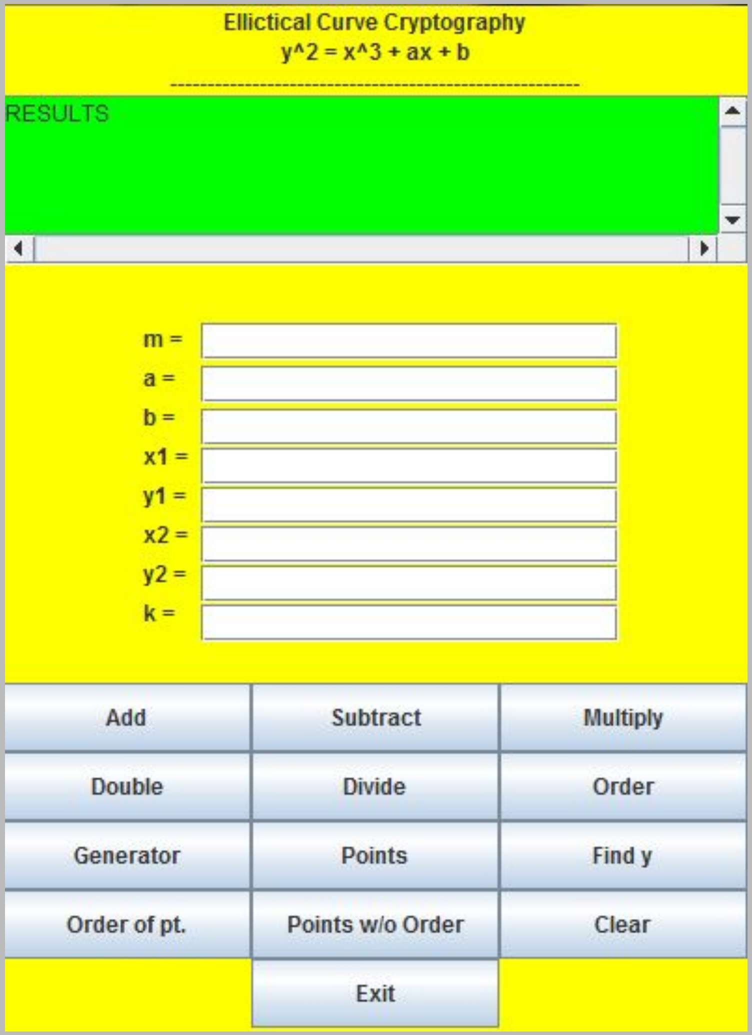
A BRIEF DESCRIPTION ABOUT THE GUI

- The GUI consists of 9 text boxes for entering different values related to the elliptic curve.They are labelled as a,b,p,x1,y1,x2,y2,order and k respectively . The notations used have their usual meanings .
- The results are displayed in a large text area which is uneditable.
- There are 12 command buttons for different operations labelled as add,sub,mull,double,Find y,Curve order,gen points,points,pts. w/o order,order of a pt. , Clear & Exit respectively .They have usual meanings and Clear serves to clear all text boxes/areas and Exit serves to Exit the application .

HOW THE GUI LOOKS



UTILITIES OF ELLIPTIC CALCULATOR



UTILITIES OF ELLIPTIC CALCULATOR

- Display Points without order
- Addition of 2 points
- Division
- Subtraction of 2 points
- Multiplication of a point with a scalar
- Doubling a point
- Display Generator points
- Display order of a point
- Order of a curve
- Find y coordinate when provided with corresponding x coordinate
- Display points with order

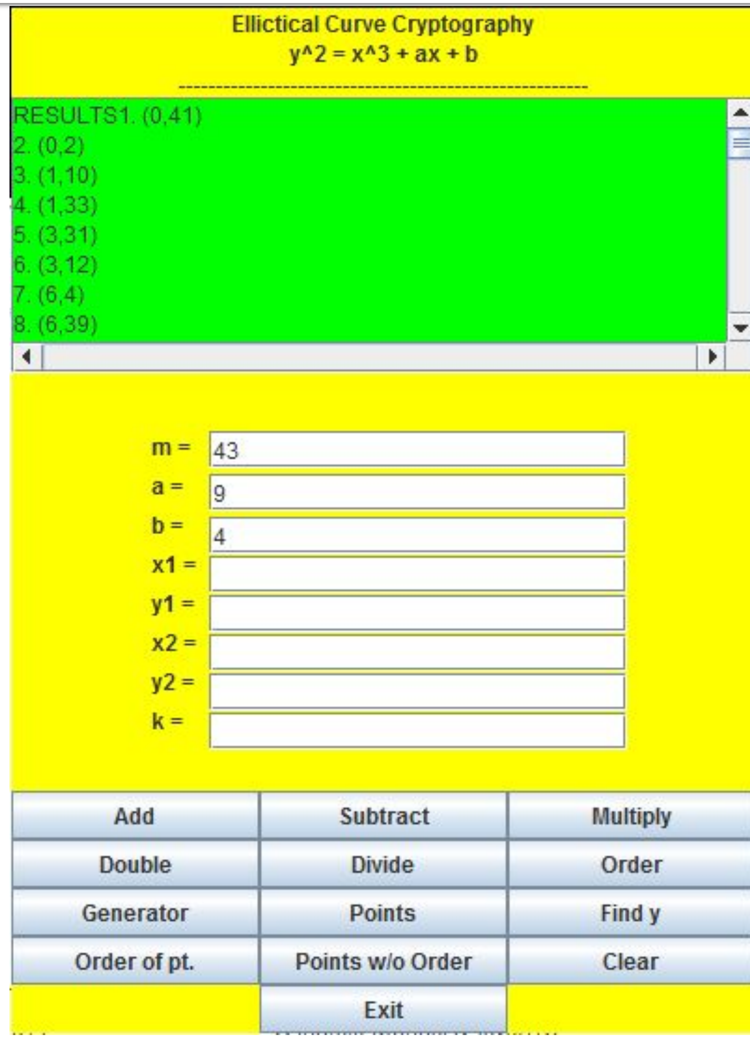
Display Points without order

- This function enables us to view those points which are included in the additive field of the elliptic curve $E_m(a,b)$.
- We need those points as inputs for calculations like Addition , Subtraction, Multiplication etc .
- In any EC Cryptosystem these points are required for the Encryption & Decryption techniques .

Algorithm of pts. w/o order

- Select p in the form of 4k+3.
- for x=0 to p-1 {
- $t1=x^3$; t2=x*a ; t3=t2+b ; xx=t3+t1 ; xx=xx mod p
- If (x=0) then y=xx; The generated pt is (x,y)
- ret=legendre(xx,p); [Calculate the Legendre symbol (*a*/*p*).It gives 1 if a is a quadratic residue module p and a!=0 (mod p)]
- If (ret=1) then {t1=p+1;t2=4;t1=t1/t2;y=xx mod $t1^p$. The generated pt is (x,y)
- y=p-y . The generated pt is (x,y)
- }

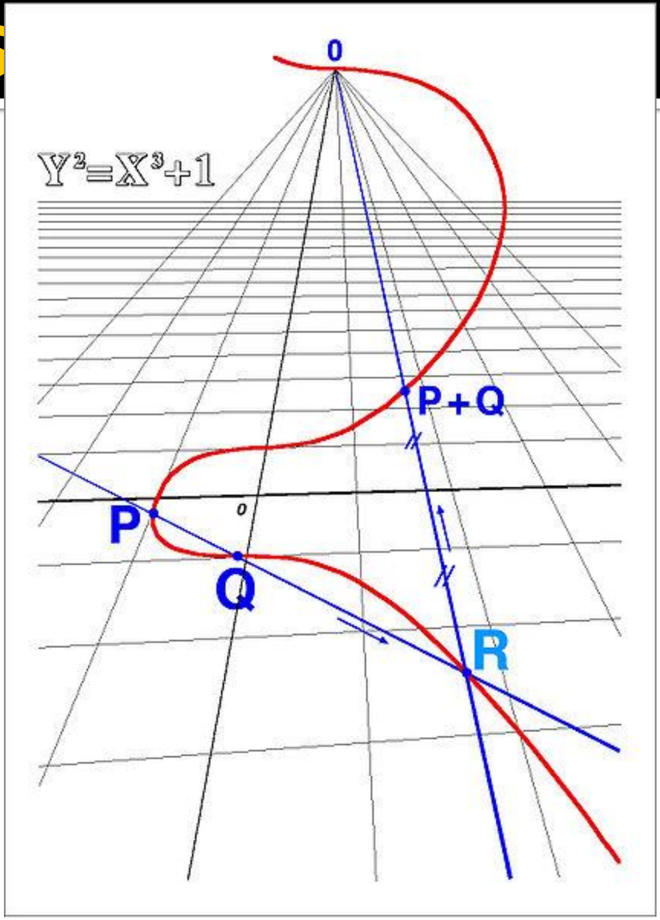
Screenshot of pts. w/o order



Addition of 2 points

For curve $E_p(a,b)$. P=prime number.

Let, $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ & $P + Q = R = (x_R, y_R)$



1st case: P != Q,

If P = -Q, R is point at infinity.

If P != -Q, $x_R = (l^2 - x_P - x_Q) \bmod p$

&

$y_R = (l(x_P - x_R) - y_P) \bmod p$

where $l = ((y_P - y_Q) / (x_P - x_Q)) \bmod p$.

2nd case: *P = Q,*

If $y_P = 0$, R is point at infinity.

If $y_P != 0$, $x_R = (l^2 - 2x_P) \bmod p$

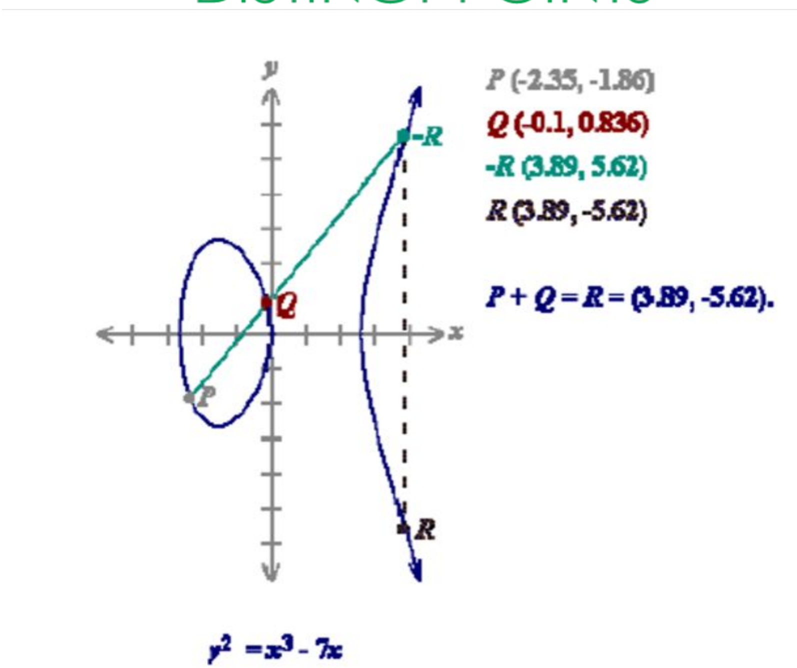
&

$y_R = (l(x_P - x_R) - y_P) \bmod p$

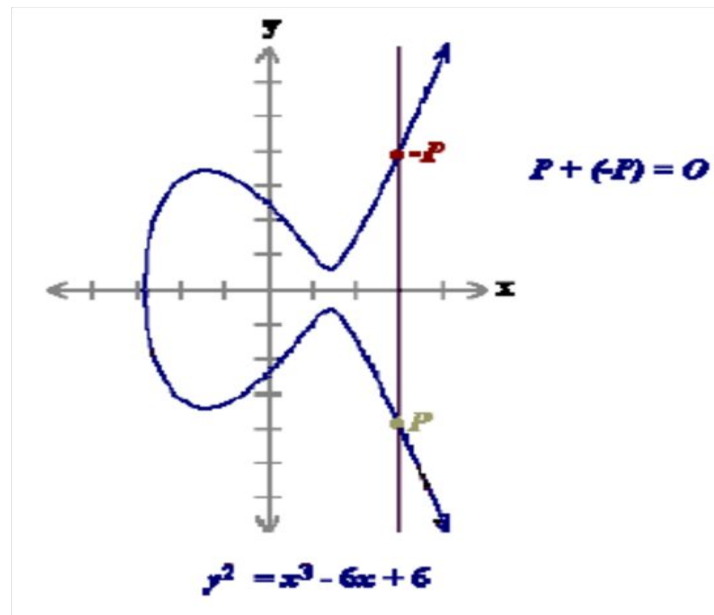
where $l = ((3x_P^2 + a)/2y_p) \bmod p$

Adding two distinct points:

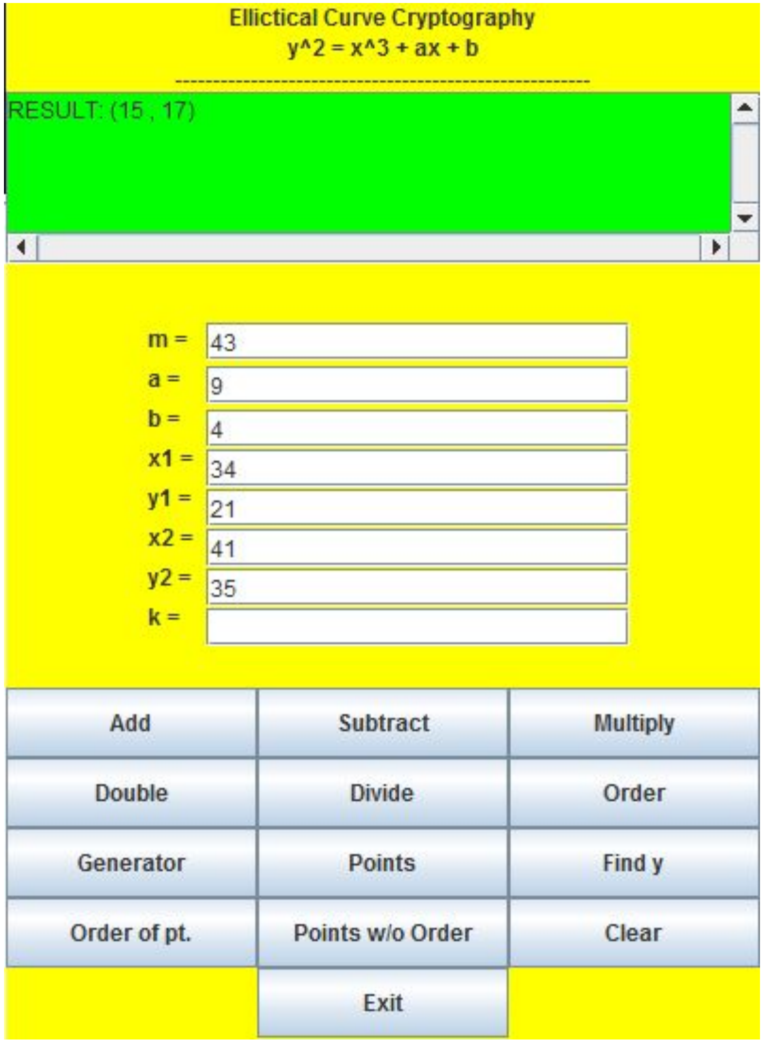
ADDING TWO DISTINCT POINTS



ADDING THE POINTS P AND –P:



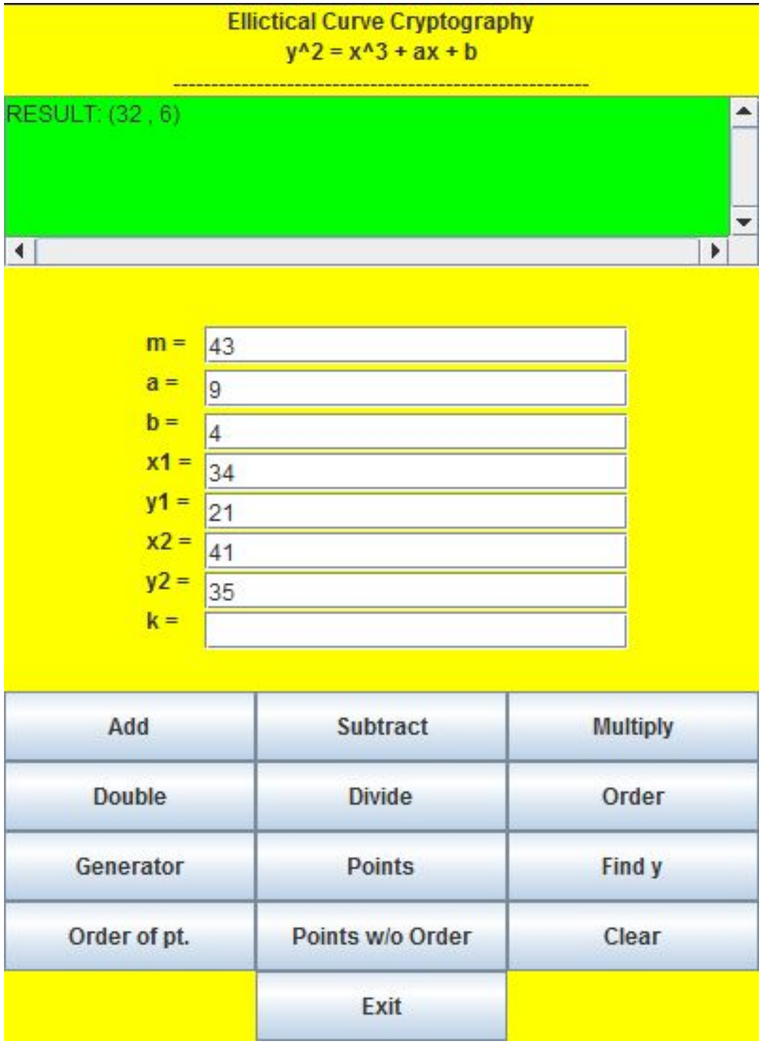
SCREENSHOT OF ADDITION



Subtraction:

- We know, (a - b) = (a + (-b)).
- If P = (x, y) is a point on elliptic curve then -P = (x, -y).
- Let, for the elliptic curve point set $E_{23}(1,0)$,
- **A) Subtracting two different points:**
- (3,10) – (12, 4) = (3, 10) + (12, -4 mod 23)
- = (3, 10) + (12, 19) = (9, 7).
- **B) Subtracting two same points**
- (3, 10) – (3, 10) = (3, 10) + (3, -10 mod 23)
- = (3, 10) + (3, 13)
- l = (13 - 10) / (3 - 3) = infinity.
- So, (3, 10) – (3, 10) = O (the point at infinity).

SCREENSHOT OF SUBTRACTION



Multiplication

Multiplication is nothing but the repeated addition.

Let, for the elliptic curve point set $E_{23}(1, 1)$,

We have to calculate 4 * (9, 5)

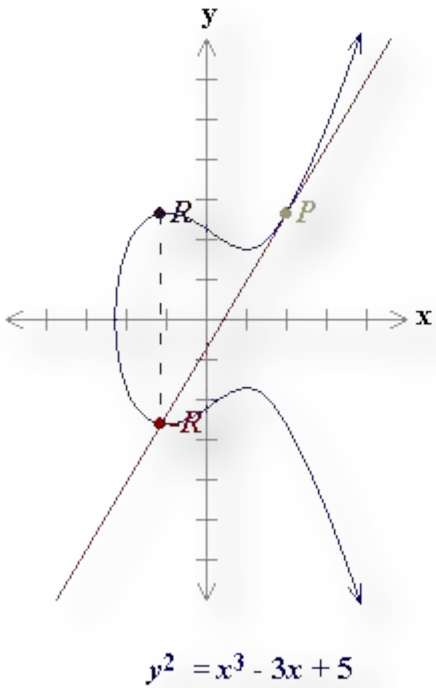
It can be expressed as (9,5) + (9,5) + (9,5) + (9,5).

Now, (2 * (9,5)) = (9,5) + (9,5) = (18,10);

Then (3 * (9,5)) = (18,10) + (9,5) = (0, 0);

Then, (4 * (9,5)) = (0,0) + (9,5) = (18,13);

GRAPHICAL EXPLANATION OF MULTIPLICATION WITH SCALAR 2



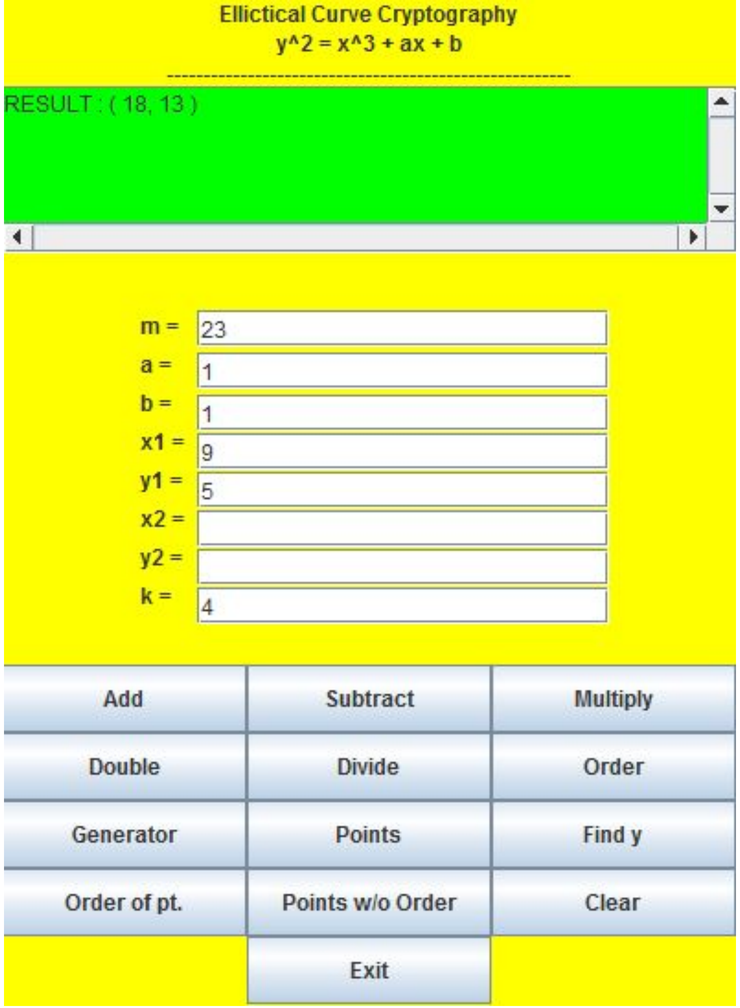
P (2, 2.65)

$-R$ (-1.11, -2.64)

R (-1.11, 2.64)

$2P = R = (-1.11, 2.64)$.

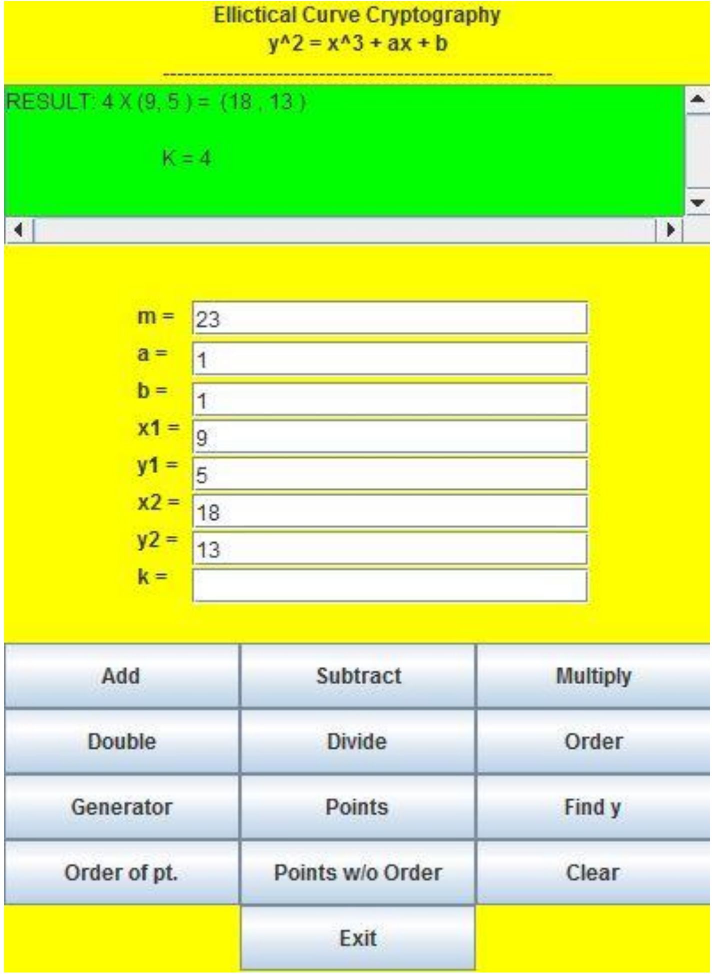
SCREENSHOT OF MULTIPLICATION



Division

- Division takes help of the Discrete logarithm problem.
- Consider the equation Q = (k * P),
- where Q, P $\in E_m(a, b)$.
- It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P. This is called discrete logarithm problem for elliptic curves.
- It determines the first k that matches.
- Only drawback is, if its very big, it takes time and result may be incorrect.

Screenshot of Division



DOUBLE OF A POINT

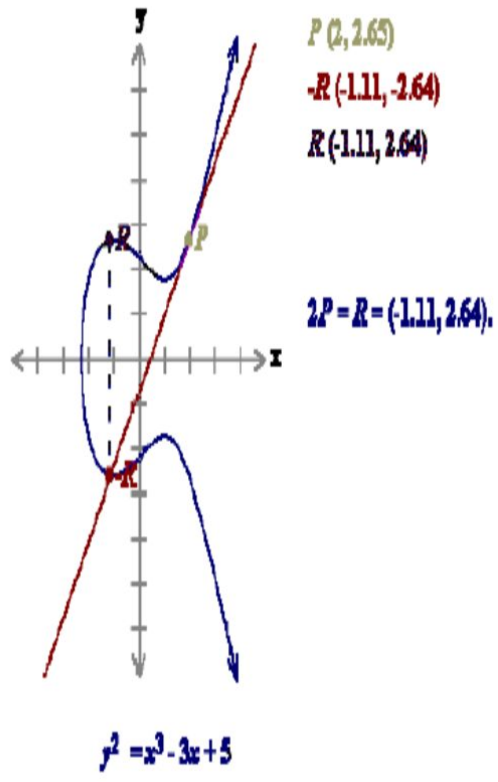
- Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve.

- . Geometrical explanation:

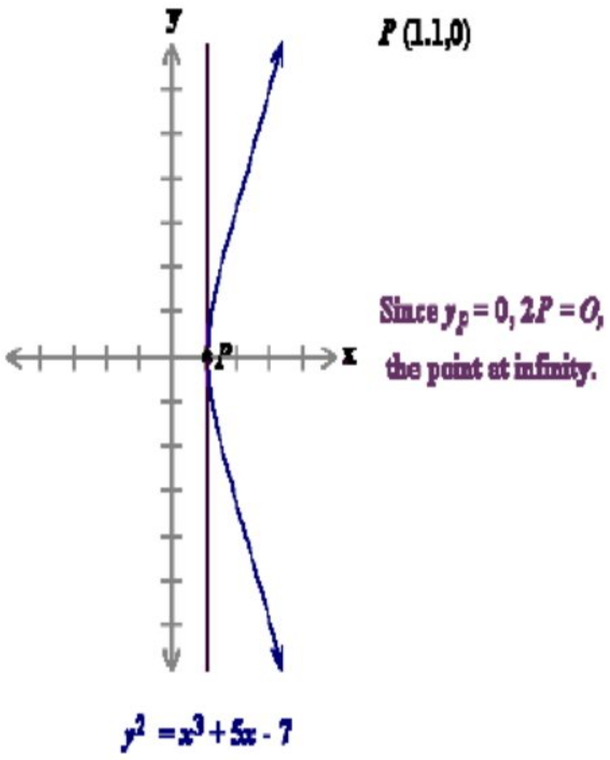
To double a point J to get L, i.e. to find L = 2J,
consider a point J on an elliptic curve as If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J.
Thus L = 2J when y_J !=0.
If y coordinate of the point J is zero then the tangent at this point intersects at a point at infinity O.
Hence 2J = O when y_J = 0.

Doubling a point:

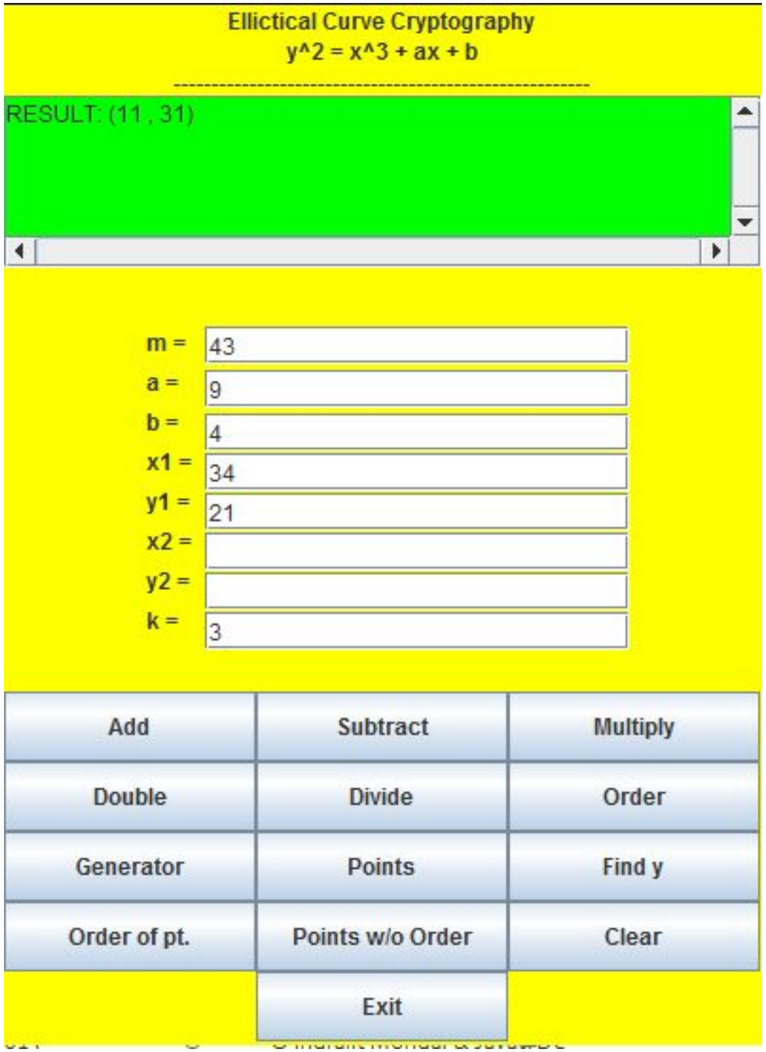
DOUBLING THE POINT P



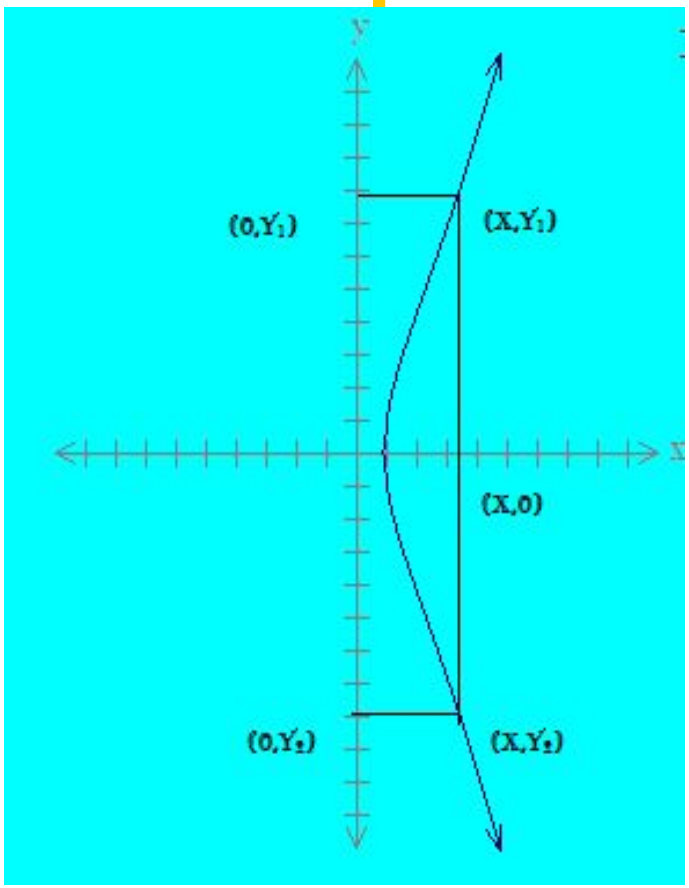
DOUBLING THE POINT



SCREENSHOT OF POINT DOUBLING

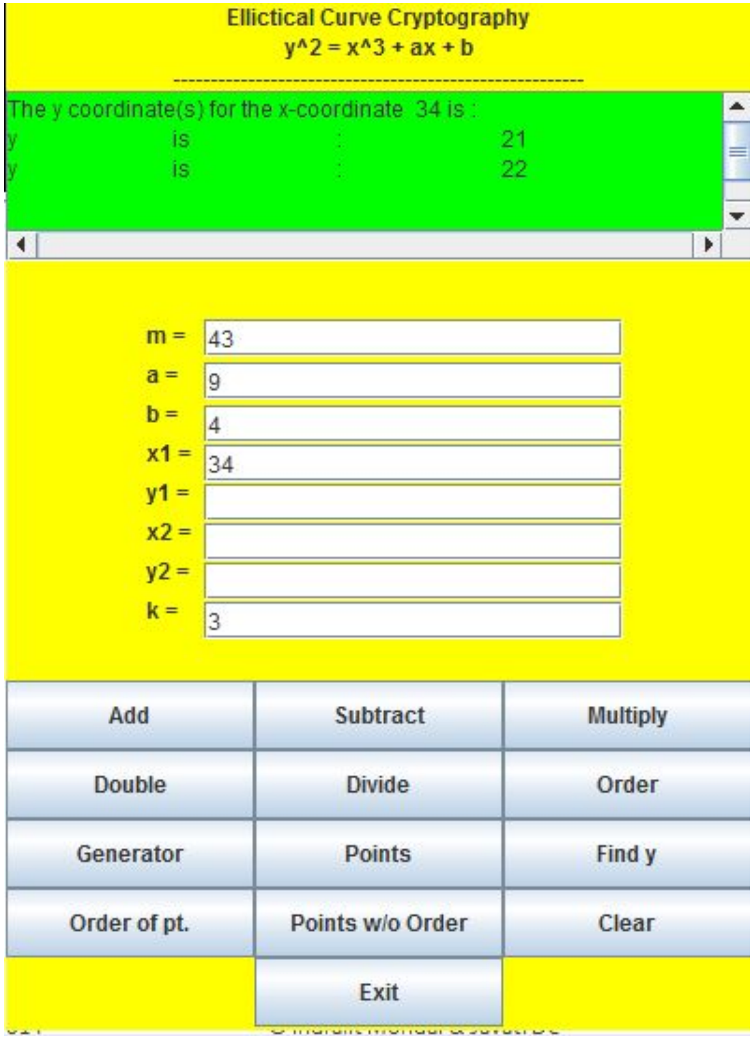


Finding y coordinate for corresponding x coordinate.



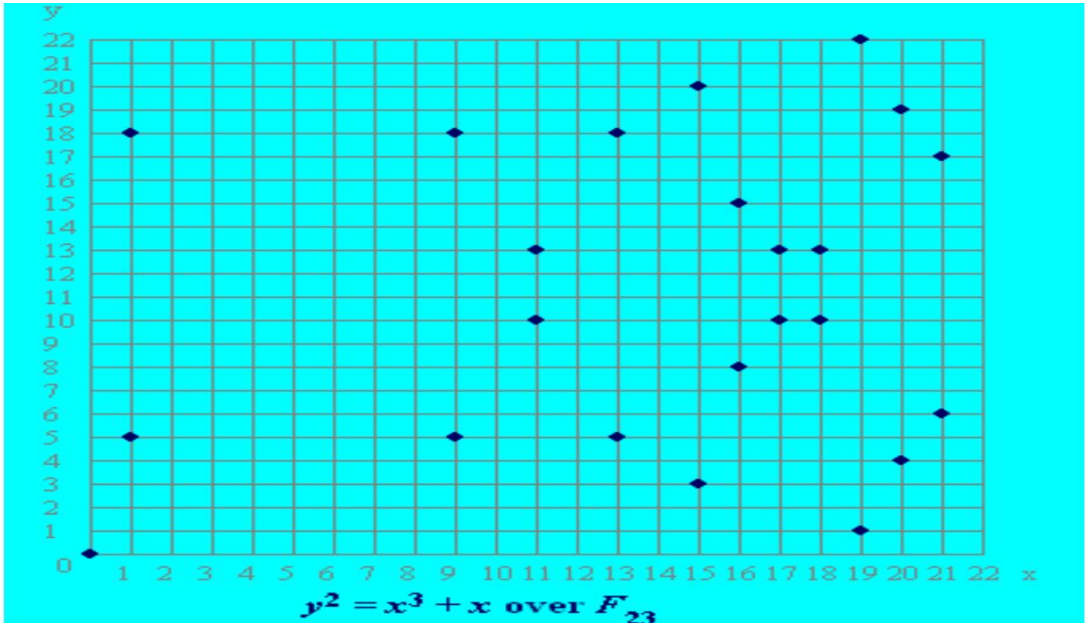
This method is all about finding the
y-coordinate of the corresponding x-coordinate
.
It is to be noted that there can be situations
where one x-coordinate corresponds
to two y-coordinates (as shown in the diagram)
.

SCREENSHOT OF FIND Y

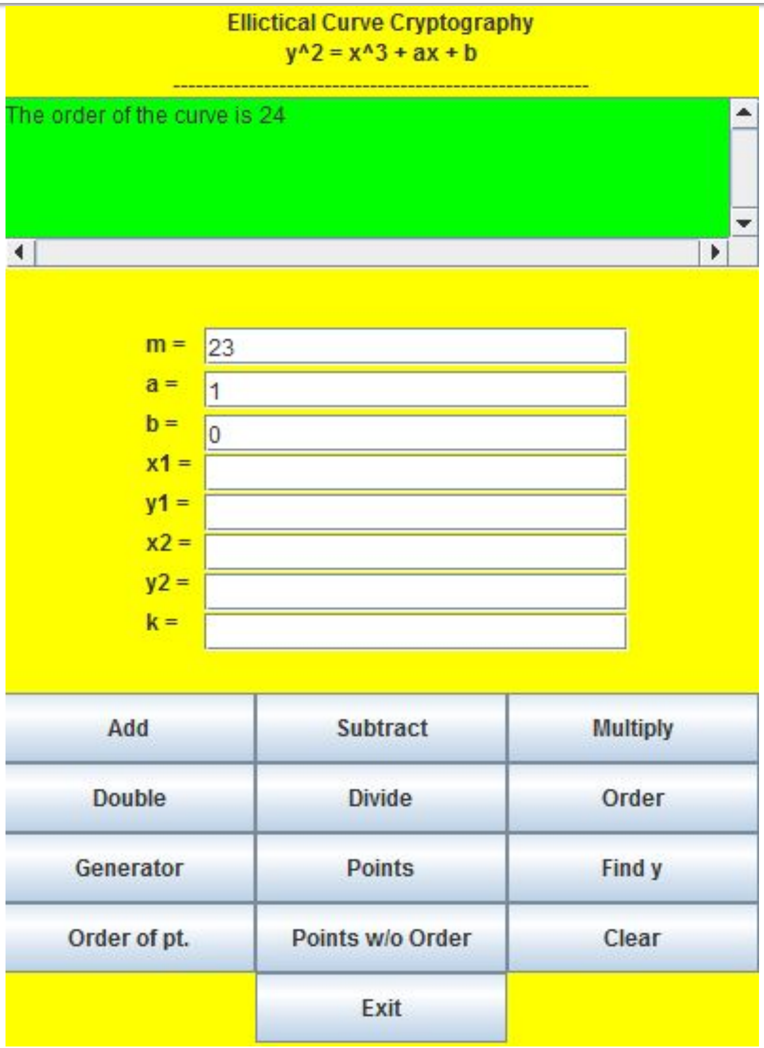


CURVE ORDER

- The 23 points which satisfy this equation are: (0,0) (1,5) (1,18)(9,5)(9,18) (11,10) (11,13) (13,5) (13,18) (15,3) (15,20) (16,8) (16,15) (17,10) (17,13) (18,10) (18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17).This total number of points is the curve order.
- These points may be graphed as below:



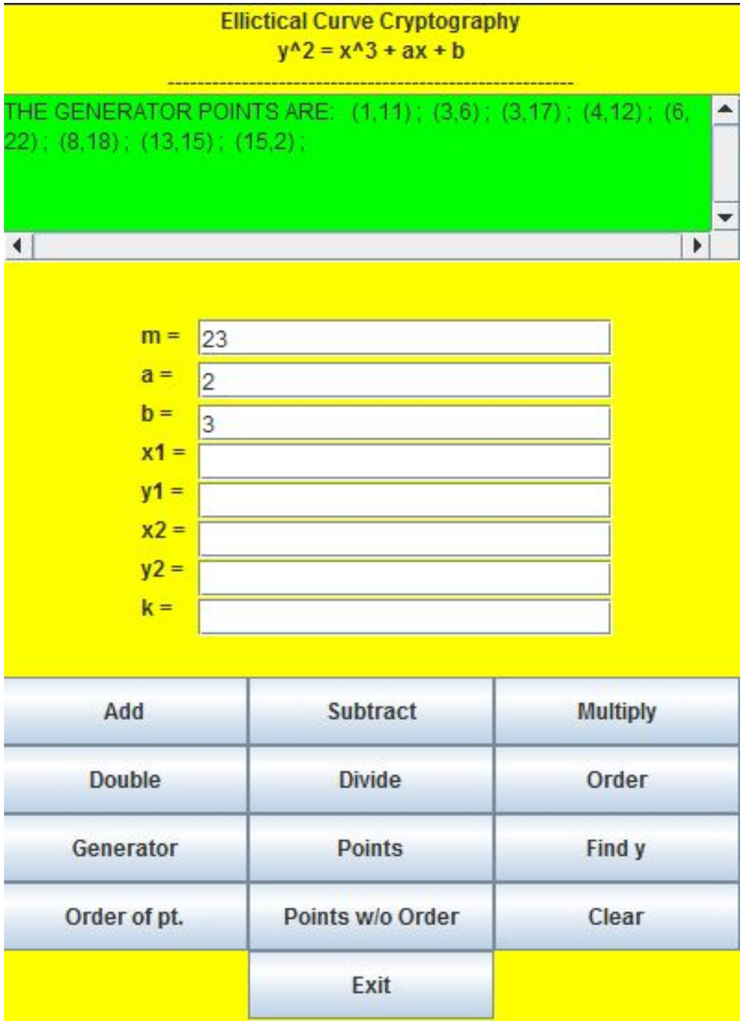
SCREENSHOT OF CURVE ORDER



GENERATOR POINTS

- There are many points in an EC . Each point has its own order . We select the highest order and demarcate those points which have the highest order as their own order.These points are the generator points.The command button “gen points” displays these points.
- It is to be noted that the highest order may be different from the curve order or may even be equal to it.
- For this operation first curve order is obtained by the “curve order” command button and then the “gen points” button is pressed .

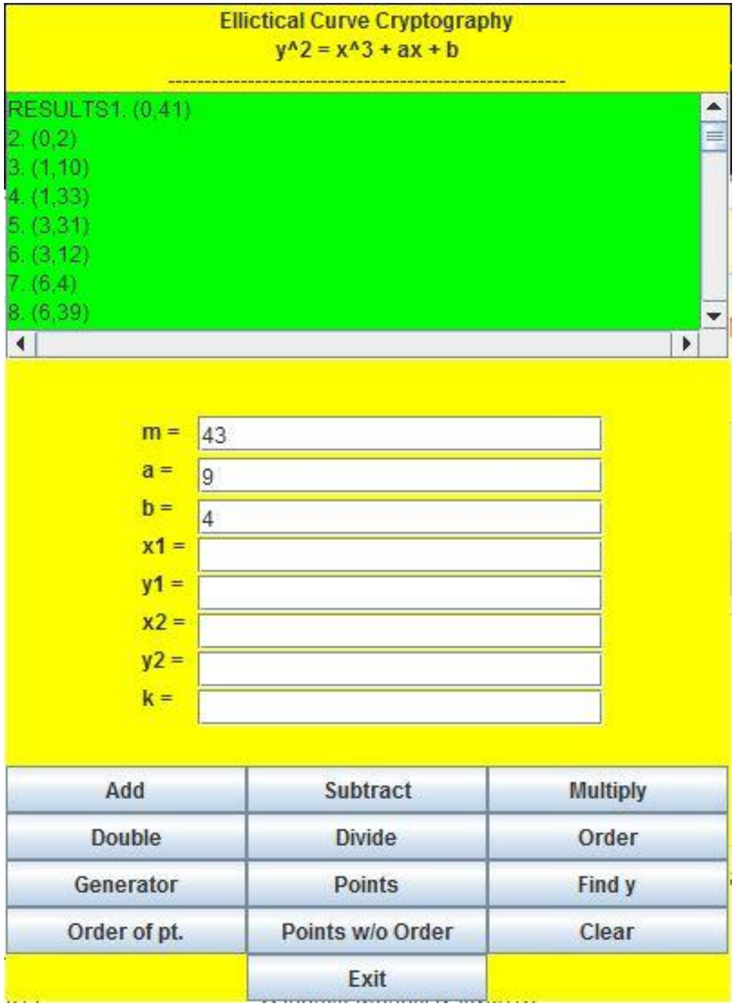
SCREENSHOT OF GENERATOR POINTS



POINTS(points with order)

- This method is baically the same as the points without order method , only that it displays the orders of the points displayed .
- As with the previous one here too first the “curve order” button is clicked to get the curve order and then “Points” button is clicked to get the points with their respective orders .

SCREENSHOT OF POINTS



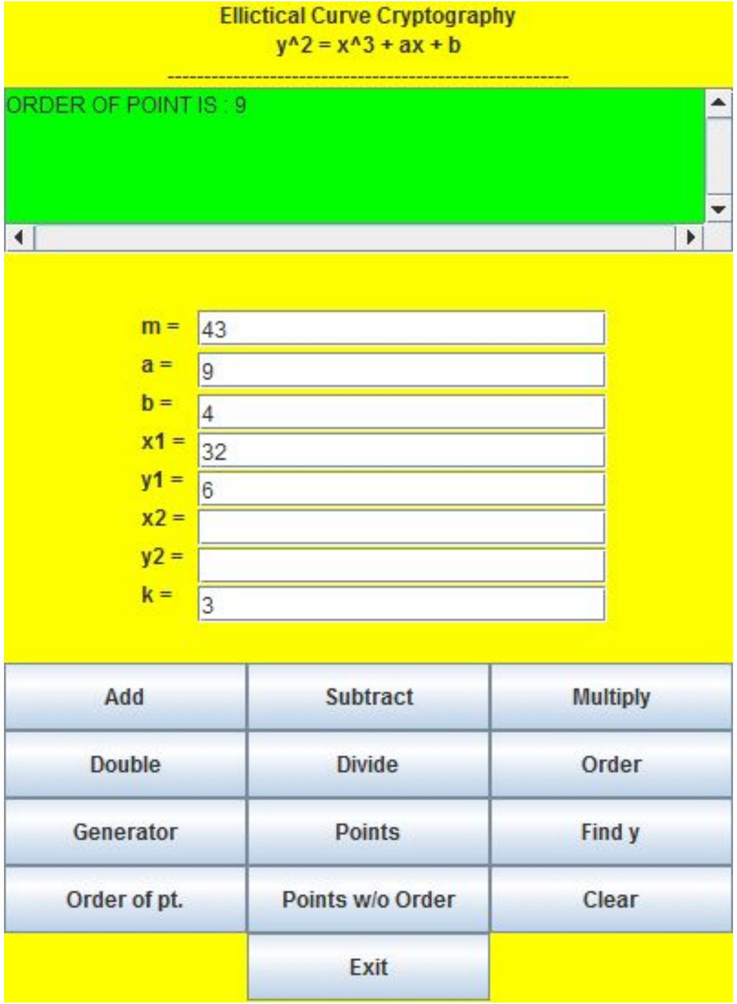
ORDER OF A POINT

- To find order of a point we basically maintain a counter & go on doubling and adding a point with itself at the same time incrementing the counter , till the value of the x-coordinate becomes 0 or infinity .

- We basically follow this algorithm :-

```
C=1;
We double the point (x1,y1) and get (xd,yd)
If(xd=0 and yd=0) then c=2;order=c;break;
C=c+1;
While(true) {
C=c+1;
Xq=xd;yq=yd;
If(xp=xd) then order=c;break;
Else
We add (x1,y1) with (xq,yq) & store the point in (xd,yd)
}
```

SCREENSHOT OF ORDER OF A POINT



Conclusion

- The “Elliptic Calculator” mainlly serves to simplify the various ECC related calculations which would otherwise become very labourious and cumbersome when performed manually.

- It provides accurate & error free results which would otherwise become erroneous when performed manually.


Thank You


Any questions?