Project Report

ELLIPTIC CALCULATOR
VERSION 3

By

Shamik Mitra (001011001029)

And

Aviroop Karmakar (001011001042)

Under the supervision and guidance of

Prof Utpal Kumar Roy

Dept. of Information Technology

Jadavpur University

Kolkata

ACKNOWLEDGEMENT

We are very happy to present before you our project titled “The Elliptic Calculator v3” . However we would not have been successful without the help of some people. Firstly we are grateful to prof. Utpal Kr. Ray without whose guidance and supervision our project would not have been possible. Next we would thank our parents whose affection and care was a constant source of inspiration for us. Next we would like to thank our dear friends whose comments and constructive criticism made us rectify various loopholes in our project.

CONTENTS

INTRODUCTION

This work describes the mathematics needed to implement Elliptic Curve Cryptography (ECC) with special attention to its implementation in Galois Field. Here the functionality of ECC, its advantages and challenges over other cryptosystems are also explained. Comparison with other cryptographic systems will also be undertaken based on aspects such as efficiency, size of the key needed to attain a certain level of security (this has implications on computational costs and time), known and probable attacks, current and predicted future attacks (based on the current growth of technology) and their prevention techniques. ECC reliability will also be looked into.

CRYPTOGRAPHY

With the rapid growth of the use of computers to exchange information electronically, the physical way of providing security by locks, sealing and signing documents and so on, is eliminated. However the need to exchange information securely is still very important and is therefore provided in electronic documents; usually by encryption and digital signatures. The science of keeping messages secure is called cryptography. Cryptography involves encryption and decryption of messages. Encryption is the process of converting a plaintext into ciphertext by using an algorithm and decryption is the process of getting back the encrypted message (Fig. 1). A cryptographic algorithm

is the mathematical function used for encryption and decryption. In addition to providing confidentiality, cryptography is often required to provide Authentication, Integrity and Non-repudiation.

The essence of cryptography is traditionally captured in the following problem: Two parties (the tradition is to call them Bob and Alice) wish to communicate over an insecure public communication channel in the presence of malevolent eavesdropper (the tradition Eve). Bob and Alice could be military jets, e-business or just friend trying to have a private conversation (Fig. 1). They can't stop Eve listening to their radio signals (or tapping their phone line, or whatever), so what can they do to keep their communication secret? One solution is for Alice and Bob to exchange a digital key, so they both know it, but it's otherwise secret.

Moreover, it is worthy to note right from the start that the hardest part of computer security is the piece between the computer and the user. While the hardest part of encryption is maintaining the security of the data when it's being entered into the keyboard and when it's being displayed on the screen. In case of the digital signatures, the hardest part is proving that the text signed is the same text that the user viewed. And finally the hardest part of

computer forensics is to know who is sitting in front of a particular computer at any time.

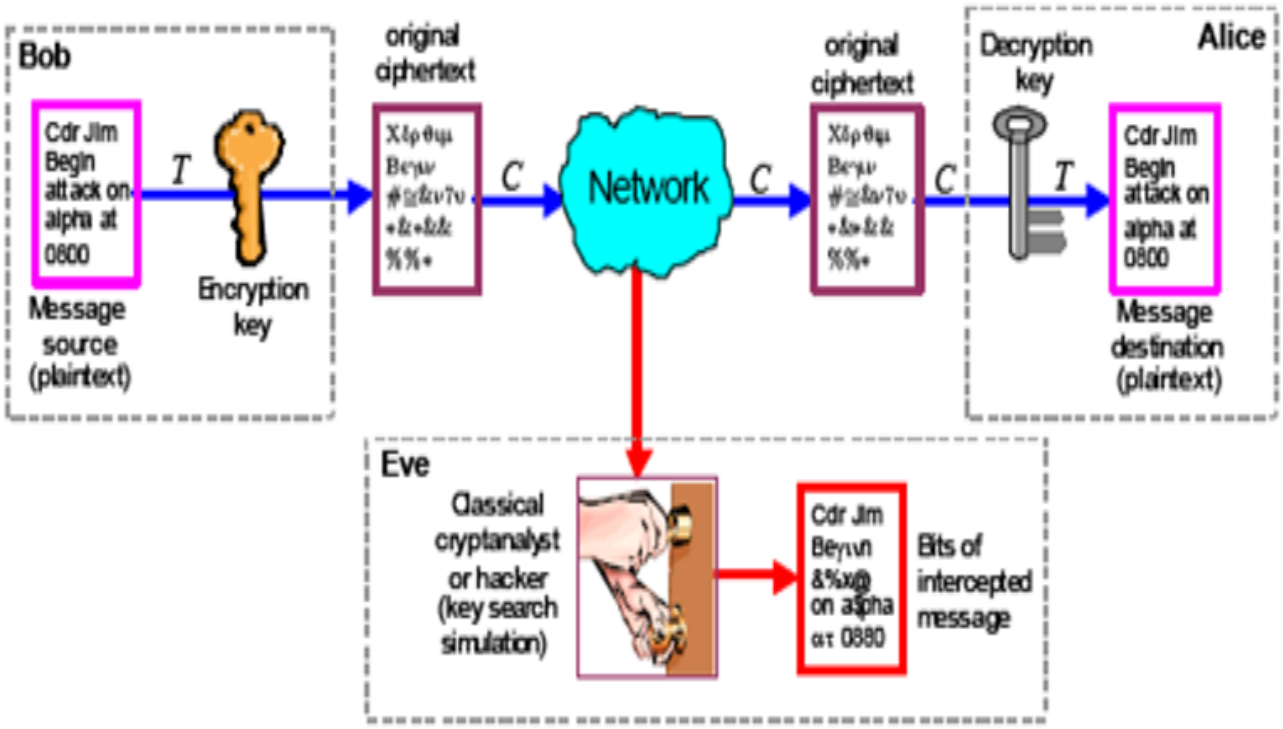


Fig. 1: Shows a schematic classical cryptographic data communication

THERE ARE TWO POPULAR KINDS OF CRYPTOGRAPHIC PROTOCOLS:

symmetric-key and asymmetric-key protocols.

In the ***symmetric-key protocol***, a common key (the secret-key) is used by both communicating partners to encrypt and decrypt messages. Among these are DES, IDEA and AES. These symmetric-key cryptosystems provide high speed key and communication but have the drawback that a common (or session) key must be established for each pair of participants. However, in 1976, Diffie and Hellman[3] introduced Public-Key Cryptography. The encoding function here is a trapdoor function-one whose inverse is impractical to implement, unless some extra information is available. This extra information (called the decrypting-key) is not required for encrypting the message, yet is essential for decrypting it in reasonable time. This makes it much easier to encrypt messages than to decrypt them. The beauty of such a system is that the encrypting process need not be kept secret. Each user has his own or a personal encrypting-function, which is public information (hence the name Public-Key) and a decoding key, which he keeps secret.

The ***public-key protocol*** employs a pair of different but associated keys. One of these keys (the public-key) is used

either for encryption (signature) of the messages and; a different key (the private-key) is used for either decryption (confidentiality) of the message. Different public-key cryptographic systems are used to provide public-key security. Among these we can mention the RSA, Diffie and Hellman (DH) key exchange algorithm, digital signature algorithm (DSA) and ElGamal cryptosystem. These systems provide these services by relying on the difficulty of different classical mathematical problems, hence provide the services in different ways.

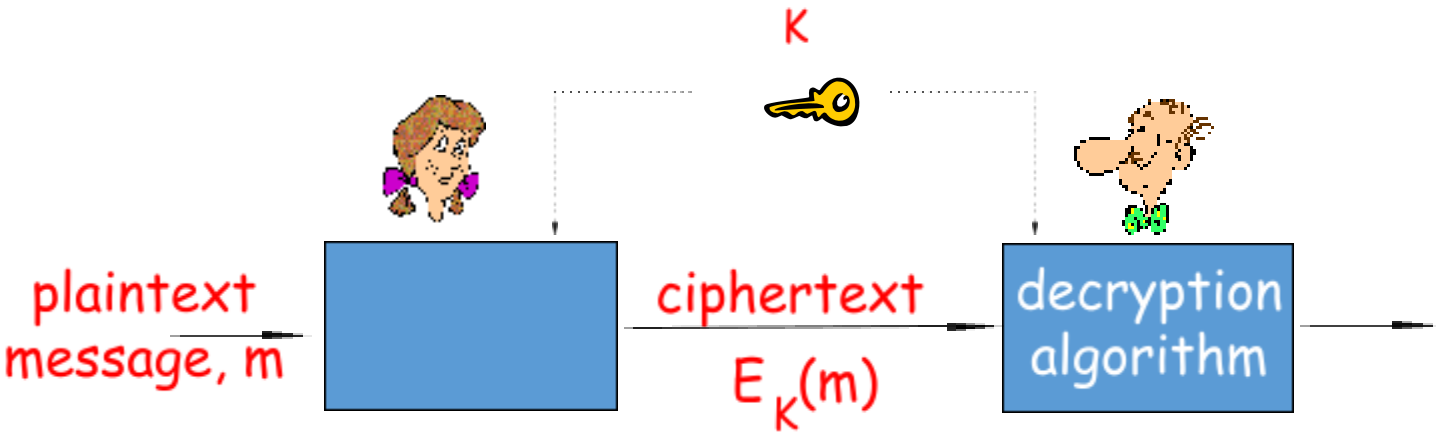
Comparison between Symmetric &Assymetric Key Cryptosystems

The public-key cryptosystems are, however, slower than the symmetric ones but provide arbitrary high levels of security and do not require an initial private key exchange between two communicating parties. In the asymmetric protocol the public-key is released to the public while the other, the private-key, is known only to its owner. Because of this feature, these cryptosystems are considered to be indispensable for secure communication and authentication over open (insecure) networks. It is designed to be computationally intractable to calculate a private-key from its associated public-key; that is, it is believed that any attempt to compute it will fail even when up-to-date technology and equipment are used. With a public-key cryptosystem, the sender can encrypt a message using the receiver's public key-without needing to know the private-key of the receiver. Therefore, they are suitable for

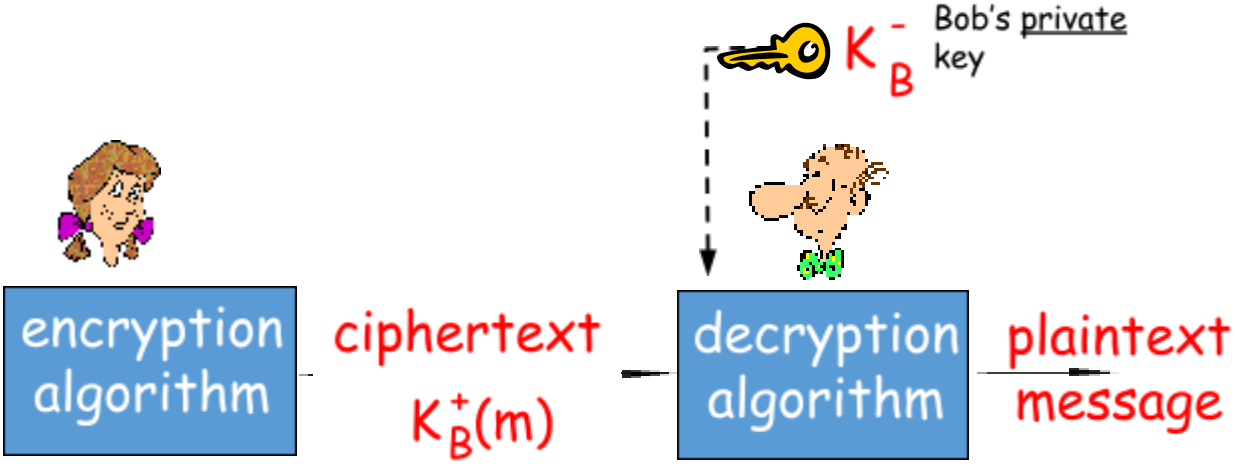
communication among the general public. Public-key cryptosystems can also be used to make a digital signature.

However, in real applications, both symmetric and asymmetric protocols are used. The public-key algorithm is first used for establishing a common symmetric-key over insecure channel. Then the symmetric system is used for secure communication with high throughput. Due to comparative slowness of the public-key algorithms, dedicated hardware is desirable for efficient implementation and operation of the cryptographic systems.

Secret key (Assymetric key) cryptosystem



Public key (Symetric key) cryptosystem



APPLICATIONS OF CRYPTOGRAPHY

Historically, cryptography was used to assure only secrecy. Wax seals, signatures, and other physical mechanisms were typically used to assure integrity of the media and authenticity of the sender. With the advent of electronic funds transfer, the applications of cryptography for integrity began to surpass its use for secrecy. Electronic cash came into being from cryptography, and the electronic credit card and debit card sprung into widespread use. The advent of public key cryptography introduced the possibility of digital signatures, and other related concepts such as electronic credentials. In the information age, cryptography has become one of the major methods for protection in all applications.

Cryptographic protocols have only recently come under intensive study, and as of this time, they are not sufficiently well developed to provide a great deal of assurance. There are several protocols that offer provable properties, primarily those designed for use with the OTP. The problem with proving properties of protocols under other schemes is that the mathematics is extremely complex for the RSA, and there is no sound mathematical basis for the DES. Much research is under way at this time in the field of protocol analysis and verification, and it is likely that once this field stabilizes, cryptographic protocols will follow suit.

Several special purpose cryptographic protocols have been developed and demonstrated sound. Most notably, the RSA key distribution protocol, a public key poker playing protocol, an OTP based dining cryptographers protocol, and the protocol used for verifying the nuclear test ban treaty .

A typical cryptographic protocol failure is encountered in the use of the RSA. It seems that if an attacker can choose the plaintext to be signed under an RSA signature system, observe the result of the signature, and then iterate the process, it is possible to get the signer to reveal the private key in a very small number of signatures (about 1 signature per bit of the key). Thus an unmodified RSA signature system requires a sound protocol to be safely used.

SECRECY IN TRANSMISSION

Most current secrecy systems for transmission use a private key system for transforming transmitted information because it is the fastest method that operates with reasonable assurance and low overhead.

If the number of communicating parties is small, key distribution is done periodically with a courier service and key maintenence is based on physical security of the keys over the period of use and destruction after new keys are distributed.

If the number of parties is large, electronic key distribution is usually used. Historically, key distribution was done with a special key-distribution-key (also known as a master-key) maintained by all parties in secrecy over a longer period of time than the keys used for a particular transaction. The "session-key" is generated at random either by one of the parties or by a trusted third party and distributed using the master-key.

The problem with master-key systems is that if the master-key is successfully attacked, the entire system collapses. Similarly, if any of the parties under a given master-key decides to attack the system, they can forge or intercept all messages throughout the entire system. Many complex private-key systems for reducing some of these problems have been proposed and used for various applications.

With the advent of public-key systems, secrecy can be maintained without a common master-key or a large number of keys. Instead, if Bob wants to communicate with Alice, Bob sends Alice a session-key encrypted with Alice's public key. Alice decrypts the session-key and uses that over the period of the transaction.

These are examples of cryptographic protocols, methods for communicating while attaining a particular cryptographic objective. These protocols are used primarily to deal with key management and system misuse problems. Many other protocols are applied to eliminate other attacks on these systems.

Secrecy in Storage

Secrecy in storage is usually maintained by a one-key system where the user provides the key to the computer at the beginning of a session, and the system then takes care of encryption and decryption throughout the course of normal use. As an example, many hardware devices are available for personal computers to automatically encrypt all information stored on disk. When the computer is turned on, the user must supply a key to the encryption hardware. The information cannot be read meaningfully without this key, so even if the disk is stolen, the information on it will not be useable.

Secrecy in storage has its problems. If the user forgets a key, all of the information encrypted with it becomes permanently unuseable. The information is only encrypted while in storage, not when in use by the user. This leaves a major hole for the attacker. If the encryption and decryption are done in software, or if the key is stored somewhere in the system, the system may be circumvented by an attacker. Backups of encrypted information are often stored in plaintext because the encryption mechanism is only applied to certain devices.

Integrity in Transmission

Many of the users of communication systems are not as much concerned about secrecy as about integrity. In an electronic funds transfer, the amount sent from one account to another is often public knowledge. What the bank cares about is that only proper transfers can take place. If an active tapper could introduce a false transfer, funds would be moved illicitly. An error in a single bit could literally cause millions of dollars to be erroneously credited or debited. Cryptographic techniques are widely used to assure that

intentional or accidental modification of transmitted information does not cause erroneous actions to take place.

A typical technique for assuring integrity is to perform a checksum of the information being transmitted and transmit the checksum in encrypted form. Once the information and encrypted checksum are received, the information is again checksummed and compared to the transmitted checksum after decryption. If the checksums agree, there is a high probability that the message is unaltered. Unfortunately, this scheme is too simple to be of practical value as it is easily forged. The problem is that the checksum of the original message is immediately apparent and a plaintext message with anidentical checksum can be easily forged. Designing strong cryptographic checksums is therefore important to the assurance of integrity in systems of this sort.

The key distribution problem in a one-key system is as before, but an interesting alternative is presented by the use of public keys. If we generate a single public-key for the entire system and throw away the private key that would go with it, we can make the checksum impossible to decrypt. In order to verify the original message, we simply generate a new checksum, encrypt with the public key, and verify that the encrypted checksum matches. This is known as a one-way function because it is hard to invert.

Actual systems of this sort use high quality cryptographic checksums and complext key distribution and maintenance protocols, but there is a trend towards the use of public keys for key maintenance.

Integrity in Storage

Integrity against random noise has been the subject of much study in the fields of fault tolerant computing and coding theory, but only recently has the need for integrity of stored information against intentional attack become a matter for cryptography.

The major mean of assuring integrity of stored information has historically been access control. Access control includes systems of locks and keys, guards, and other mechanisms of a physical or logical nature. The recent advent of computer viruses has changed

this to a significant degree, and the use of cryptographic checksums for assuring the integrity of stored information is now becoming widespread.

As in the case of integrity in transmission, a cryptographic checksum is produced and compared to expectations, but storage media tends to have different properties than transmission media. Transmitted information is typically more widely available over a shorter period of time, used for a relatively low volume of information, and accessed at a slower rate than stored information. Thees parameters cause different tradeoffs in how cryptosystems are used.

Authentication of Identity

Authenticating the identity of individuals or systems to each other has been a problem for a very long time. Simple passwords have been used for thousands of years to prove identity. More complex protocols such as sequences of keywords exchanged between sets of parties are often shown in the movies or on television. Cryptography is closely linked to the theory and practice of using passwords, and modern systems often use strong cryptographic transforms in conjunction with physical properties of individuals and shared secrets to provide highly reliable authentication of identity.

Determining good passwords falls into the field known as key selection. In essence, a password can be thought of as a key to a cryptosystem that allows encryption and decryption of everything that the password allows access to. In fact, password systems have been implemented in exactly this way in some commercial products.

The selection of keys has historically been a cause of cryptosystem failure. Although we know from Shannon that H(K) is maximized for a key chosen with an equal probability of each possible value (i.e. at random), in practice when people choose keys, they choose them to make them easy to remember, and therefore not at random. This is most dramatically demonstrated in the poor selection that people make of passwords.

On many systems, passwords are stored in encrypted form with read access available to all so that programs wishing to check passwords

needn't be run by privileged users. A side benefit is that the plaintext passwords don't appear anywhere in the system, so an accidental leak of information doesn't compromise system wide protection.

A typical algorithm for transforming any string into an encrypted password is designed so that it takes 10 or more msec/transformation to encode a string. By simple calculation, if only capital letters were allowed in a password, it would take .26 seconds to check all the one letter passwords, 6.76 seconds to check all the 2 letter passwords, 4570 seconds for the 4 letter passwords, and by the time we got to 8 letter passwords, it would take about 2*10**9 seconds (24169 days, over 66 years).

For passwords allowing lower case letters, numbers, and special symbols, this goes up considerably. Studies over the years have consistently indicated that key selection by those without a knowledge of protection is very poor. In a recent study , 21% of the users on a computer system had 1 character passwords, with up to 85% having passwords of 1/2 the maximum allowable length, and 92% having passwords of 4 characters or less. These results are quite typical, and dramatically demonstrate that 92% of all passwords could be guessed on a typical system in just over an hour.

Several suggestions for getting unpredictable uniform random numbers include the use of low order bits of Geiger counter counts, the use of the time between entries at a keyboard, low order bits of the amount of light in a room as measured by a light sensitive diode, noisy diode output, the last digit of the first phone number on a given page of a telephone book, and digits from transcendental numbers such as Pi.

Credentialing Systems

A credential is typically a document that introduces one party to another by referencing a commonly known trusted party. For example, when credit is applied for, references are usualy requested. The credit of the references is checked and they are contacted to determine the creditworthiness of the applicant. Credit cards are often used to credential an individual to attain further credit cards. A driver's license is a form of credential, as is a passport.

Electronic credentials are designed to allow the credence of a claim to be verified electronically. Although no purely electronic credentialing systems are in widespread use at this time, many such systems are being integrated into the smart-card systems in widespread use in Europe. A smart-card is simply a credit-card shaped computer that performs cryptographic functions and stores secret information. When used in conjunction with other devices and systems, it allows a wide variety of cryptographic applications to be performed with relative ease of use to the consumer.

Electronic Signatures

Electronic signatures, like their physical counterparts, are a means of providing a legally binding transaction between two or more parties. To be as useful as a physical signature, electronic signatures must be at least as hard to forge, at least as easy to use, and accepted in a court of law as binding upon all parties to the transaction.

The need for these electronic signatures is especially accute in business dealings wherein the parties to a contract are not in the same physical vacinity. For example, in an international sale of an airplane, signatures are typically required from two bankers, two companies, two insurance agencies, two attorneys, two governments, and often several other parties. The contracts are hundreds of pages long, and signatures must be attained within a relatively shoert period of time on the same physical document. Faxcimily signatures are not legally binding in all jurisdictions, and the sheer length of a document precludes all parties reading the current copy as they meet at the table to affix signatures. Under current law, all parties must meet in one location in order to complete the transaction. In a transatlantic sale, $100,000 in costs can easily be encurred in such a meeting.

An effort in Europe is currently underway to replace physical signatures with electronic signatures based on the RSA cryptosystem. If this effort succeeds, it will allow many millions of dollars to be saved, and launch the era of digital signatures into full scale motion. It will also create a large savings for those who use the system, and therefore act to force others to participate in order to remain competitive.

Electronic Cash

There are patents under force throughout the world today to allow electronic information to replace cash money for financial transactions between individuals. Such a system involves using cryptography to keep the assets of nations in electronic form. Clearly the ability to forge such a system would allow national economies to be destroyed in an instant. The pressure for integrity in such a system is staggaring.

Threshold Systems

Thresholding systems are systems designed to allow use only if a minimal number of parties agree to said use. For example, in a nuclear arms situation, you might want a system wherein three out of five members of the joint chiefs of staff agree. In a banking situation, a safe might only be openned if 4 out of the authorized 23 people allowed to open the safe were present. Such systems preclude a single individual acting alone, while allowing many of the parties to a transaction to be absent without the transaction being halted.

Most threshold systems are based on encryption with keys which are distributed in parts. The most common technique for partitioning a key into parts is to form the key as the solution to N equations in N unknowns. If N independent equations are known, the key can be determined by solving the simultaneous equations. If less than N equations are known, the key can be any value since there is still an independent variable in the equations. Any number can be chosen for N and equations can be held by separate individuals. The same general concept can be used to form arbitrary combinations of key requirements by forming ORs and ANDs of encryptions using different sets of keys for different combinations of key holders _. The major difficulties with such a system lie in the key distribution problem and the large number of keys necessary to achieve arbitrary key holder combinations.

Systems Using Changing Keys

Shannon has shown us that given enough reuse of a key, it can eventually be determined. It is thus common practice to regularly change keys to limit the exposure due to successful attack on any given key. A common misconception is that changing a key much

more often than the average time required to break the cryptosystem, provides an increased margin of safety.

If we assume the key is chosen at random, and that the attacker can check a given percentage of the keys before a key change is made, it is only a matter of time before one of the keys checked by the attacker happens to correspond to one of the random keys. If the attacker chooses keys to attack at random without replacement over the period of key usage, and begins again at the beginning of each period, it is 50% likely that a currently valid key will be found by the time required to try 50% of the total number of keys, regardless of key changes. Thus if a PC could try all the DES keys in 10 years, it would be 50% likely that a successful attack could be launched in 5 years of effort. The real benefit of key changes is that the time over which a broken key is useful is limited to the time till the next key change. This is called limiting the exposure from a stolen key.

Hardware to Support Cryptography

Historically, cryptography has been carried out through the use of cryptographic devices. The use of these devices derives from the difficulty in performing cryptographic transforms manually, the severe nature of errors that result from the lack of redundancy in many cryptographic systems, and the need to make the breaking of codes computationally difficult.

In WWII, the ENIGMA machine was used by the Germans to encode messages, and one of the first computers ever built was the BOMB, which was designed to break ENIGMA cryptograms. Modern supercomputers are used primarily by the NSA to achieve the computational advantage necessary to break many modern cryptosystems. The CRAY could be easily used to break most password enciphering systems, RSA systems with keys of length under about 80 (circa 1986) are seriously threatened by the CRAY, and even the DES can be attacked by using special purpose computer hardware.

ELLIPTIC CURVE CRYPTOGRAPHY

In the mid 1980s researchers noticed that another source of hard problems might be discovered by looking at the elliptic curves. The invention of Elliptic Curve Cryptography (ECC) offered a new level of security for public key cryptosystemswhich provide both encryption and digital signatures services. One potential use of elliptic curves is in the definition of public-key cryptosystems that are close analogs of existing schemes like RSA, ElGamal, DSA and DH etc. Furthermore, elliptic curve can provide versions of public-key methods that, in some cases, are faster and use smaller keys, while providing an equivalent level of security. Their advantage comes from using different kind of mathematical group for public-key arithmetic.

To date many research papers in Elliptic Curve Cryptography (ECC) have been published by researchers all over the world, as can be viewed in the refs. However, the idea of using elliptic curves in cryptography is still considered a difficult concept and is neither widely accepted nor understood by typical technical people. The problem may stem from the fact that there is a large gap between the theoretical mathematics of elliptic curves and the applications of elliptic curves in cryptography.

Introduction and history of elliptic curve: The name “elliptic curve” is based on the ellipse. Elliptic curves were first discovered after the 17th century in the form of Diophantine equation, $y2-x3 = c$, for $c \in R$. Further, it is important to note that, however, it is easy to calculate the surface of the ellipse, it is hard to calculate the circumference of the ellipse. The calculation can be reduced to an integral:

$$\int \frac{1}{\sqrt{x^3 + Ax + B}}\,dx \qquad (1)$$

This integral, which cannot be solved easily, was the reason to consider the curve Y2 = X3+AX+B. This was done already during the 18th century. Probably, it was Abel in ca. 1820 who introduced the addition of points on the curve. At the moment there are several definitions for an elliptic curve. However, the following definition is used usually:

DEFINITION:An elliptic curve E, defined over an arbitrary field K, is a non-supersingular plain projective third degree curve over K with a K-rational point O (i.e., with coordinates in K) over curve E.

Such a curve can be described by its so-called Weierstraβ form, in homogeneous coordinates x,y,z:

$$E: y^2z+a_1xyz+a_3yz^2 = x^3+a_2x^2z+a_4xz^2+a_6z^3 \quad \text{--- (2)}$$

where, a1 ,...., a6 ε K, such that the discriminant Δ ≠ 0. This discriminant is a polynomial expression in the coefficient a1 ,...., a6. The restriction Δ ≠ 0 is necessary and sufficient for E in order to be a non-singular. The curve E has exactly one K-rational point at 'infinity', i.e., z = 0, the point (0 : 1 : 0). This point plays the role of O (origin). Sometimes we want to express that a curve E is based over field K. We do this by notation E/K.

In general, it will restricted ourselves only to the affine part $z \neq 0$ of elliptic curves E:

$$E: y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6 \quad \text{----------} \quad (3)$$

For fields K of characteristic >2 such as Fp with p>3 we can transform the Weierstraβ form for the affine curve by the coordinate transform:

$$\chi = x + \frac{a_1^2+4a_2}{12} \quad \text{and} \quad y = y + \frac{a_1}{2}x + \frac{a_3}{2} \quad \text{----------} \quad (4)$$

to a curve of the form: E/K: $Y^2 = X^3+aX+b$

where, a,b ε K such that the discriminant $\Delta = 4a^3+27b^2 \neq 0$. This form also known as Weierstraβ short form will be used later.

Remark: For practical applications finite field of the form are very important. For such elliptic curves the theory as mentioned above has to be modified.

In 1955, Yutaka Taniyama asked some questions about elliptic curves, i.e., curves of the form $y^2 = x^3+ax+b$ for constants a and b. Hellegouarch studied the application of elliptic curves for solving Fermat's Last Theorem in 1971. Elliptic curves can also be looked at as mathematical constructions from cryptography.

Elliptic curve cryptosystem (ECC) is relatively new. The ECC was first introduced by Miller and independently by Koblitz in the mid 1980s and today it has evolved into a mature public-key cryptosystem. It was also recently endorsed by the U.S. government . The ECC from the very beginning was proposed as an alternative

to established public-key systems such as DH, DSA, RSA and ElGamal cryptosystems. This is so, because elliptic curves do not introduce new cryptographic algorithms, but they implement existing public-key algorithms using elliptic curves. In this way, variants of existing schemes can be devised that rely for their security on a different underlyinghard problem.

ADVANTAGES OF USING ECC:

Summarizing the above discussions, following advantages can be derived.

• Size of the key : The size of the key required for encryption and digital signatures is surely far less than other systems.

• Less time for encryption.

•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.

• None of the currently known algorithms for the solution of the DL problem can be applied.The use of brute force attack on ECC takes a lot longer time to be successful.

• Any cryptographic scheme/protocol based on discrete logarithms can be easily converted to elliptic curve form.

• If the order of the fixed point F is an n-bit prime, then computing k from k*F and F takes roughly 2^(n/2) operations.

DISADVANTAGES OF USING ECC :

• ECC utilizes elliptic curves, generators and finite fields, and points of a curve, which can lead to more complex calculations, that could strain an embedded processor.

• This can be circumvented with lookup tables for elliptic curves, but this can eat up valuable resources on handheld and portable devices.

• The field has not been tested widely, compared to other fields.

• Probability of future discovery of subexponential attack. A successful discovery of an attack on the system can weaken the immunity of the system.

• ECC systems are slower than RSA in public key operations. So in applications requiring vast public key encryptions, the system is not desirable.

Applications of ECC :

Applications involving Constrained Channels : constrained channels are those, which are limited in memory, processing and other resources. In such situations, following advantages of ECC are surely more helpful :-

• Shorter keys

• Shorter signatures

• Shorter certificates

• Simple generation of key pair

Use of Smart Cards and Tokens : implementations of smart cards require use of cryptography. Implementing ECC in such cases is indispensable.

ECC Review :

Elliptic curve cryptography has proven to be a viable solution for the implementation of public-key cryptosystems. As widespread use of the Internet continues to increase, transferring data over this insecure medium has become a significant issue. With smaller key sizes and lower processing requirements than other public key cryptosystems, elliptic curve cryptography lends itself well to sending information securely over the Internet where bandwidth and processing capabilities are limited.

The system may be the next generation of public key technique. This field still remains a rich field of research and development, there being many stones yet unturned.

With more advancements in the field, the system might completely replace RSA & other techniques or at least act as the other solution.

Elliptic Curve Group Over Real Field:

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

Arithmetic Over Real Field(Geometric approach):

- 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.

MATHEMATICS OF ECC :-

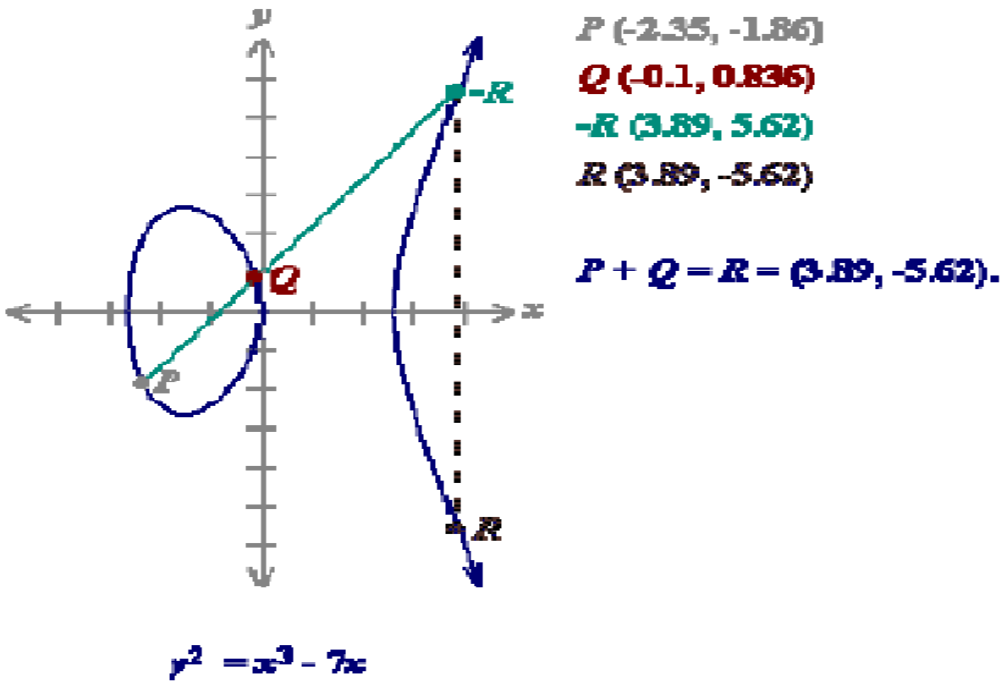
Now the mathetical tools of ECC which are relevant to our project would be discussed .

- **Adding two points:**

  Adding distinct points P and Q

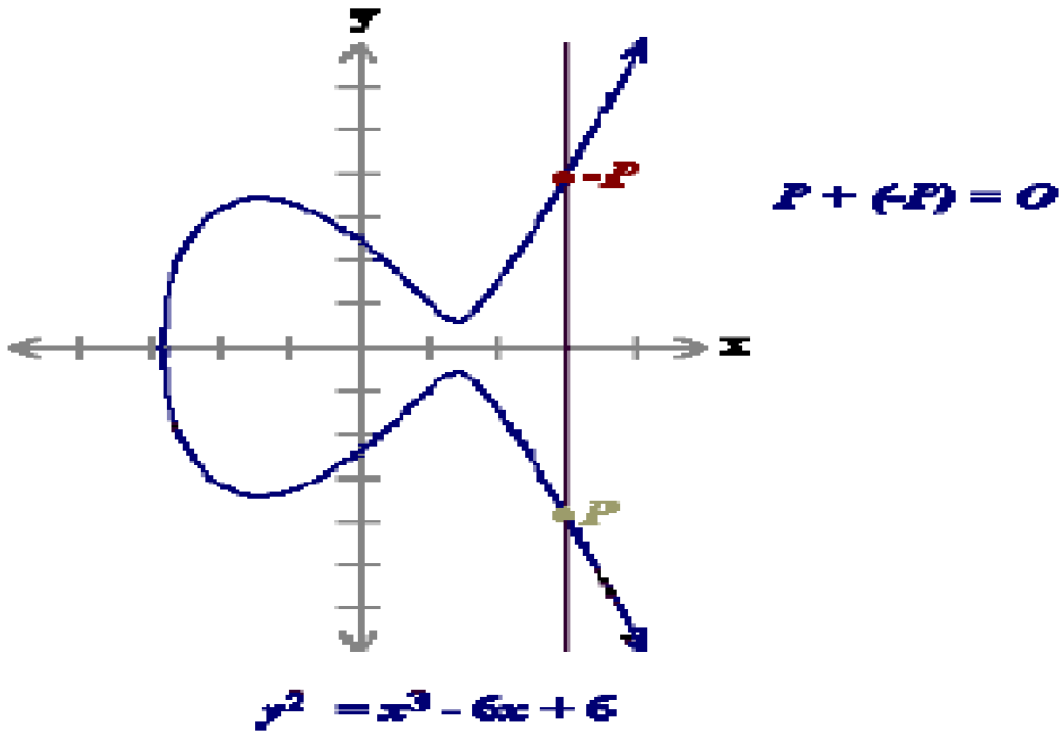
  Suppose that P and Q are two distinct points on an elliptic curve, and the P is not -Q. To add the points P and Q, a line is drawn through the two points. This line will intersect the elliptic curve in exactly one more point, call -R. The point -R is reflected in the x-axis to the point R. The law for addition in an elliptic curve group is P + Q = R.

  For example:



ADDING THE POINTS P AND –P

The line through P and -P is a vertical line which does not intersect the elliptic curve at a third point; thus the points P and -P cannot be added as previously. It is for this reason that the elliptic curve group includes the point at infinity O. By definition, P + (-P) = O. As a result of this equation, P + O = P in the elliptic curve group . O is called the additive identity of the elliptic curve group; all elliptic curves have an additive identity. For example :-



Algebraic Approach:

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, $x_R = l^2 - x_P - x_Q$ & $y_R = l(x_P - x_R) - y_P$

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

2nd case: P = Q,

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

If $y_P != 0$, $x_R = l^2 - 2x_P$ & $y_R = l(x_P - x_R) - y_P$

where $l = (3x_P^2 + a)/2y_p$.

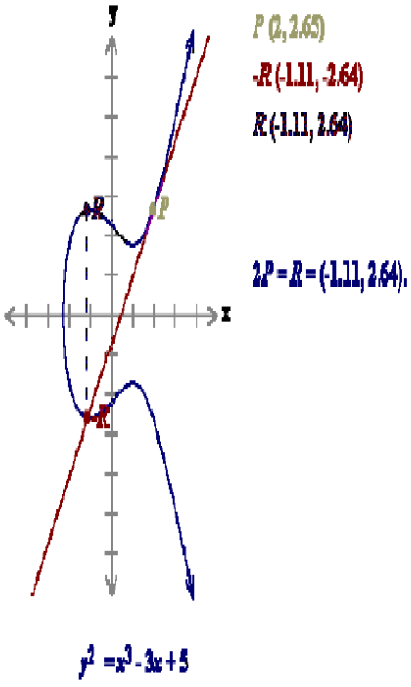
Multiplication can be expressed as repeated addition.

Exp: 4P = P + P + P + P.

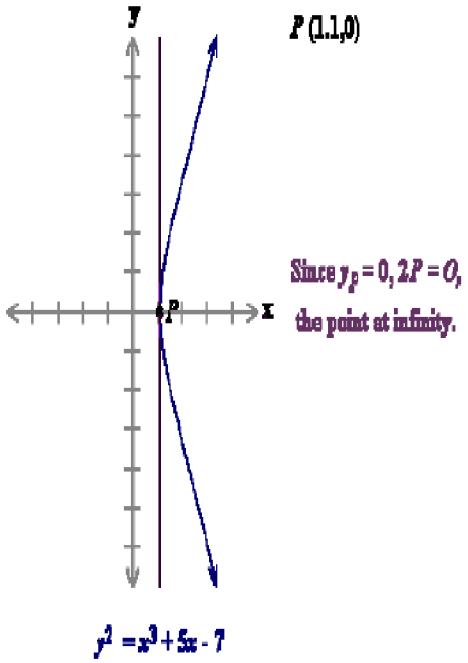
Doubling a point:

To add a point P to itself, a tangent line to the curve is drawn at the point P. If yP is not 0, then the tangent line intersects the elliptic curve at exactly one other point, -R. -R is reflected in the x-axis to R. This operation is called doubling the point P; the law for doubling a point on an elliptic curve group is defined by:

P + P = 2P = R.



DOUBLING THE POINT P



DOUBLING THE POINT P IF $Y_P = 0$

P+O=P, 2P=0. 3P=P, 4P=O

Elliptic Key Cryptography Mechanism:

Key Generation

- Decide on a, b, m. This gives the Elliptic Curve $E_m(a,b)$.
- Find the generator point G on elliptic group $E_m(a,b)$.
- Select an integer n_B as private key.
- The public key is $P_B = n_B * G$.
- The information on public domain $\{ E_m(a,b), P_B \}$.
- The information on private domain $\{ n_B \}$.

Encryption Technique:

The cipher text C_m is consisted of the following pair of points.

$$C_m = [(k * G), \{ P_m + (k * P_B)\}]$$

Where, k = sender selected positive random number,

G = generator point,

P_m = plaintext in elliptic curve point format.

P_B = receiver's public key

k = random number chosen by the sender

Decryption Technique:

$(P_m + k P_B) – n_B(k * G) = P_m + k (n_B*G) – n_B(k *G)$

$= P_m$ (recovered plaintext)

Generated Output:

Cipher text file.

Security Of ECC:

Elliptic Curve 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).

Bruit 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.

Bruit-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.

Problem Design :

Next we go straight to the details of our project work . We have created an application called “Elliptic Calculator” , the next modules is dedicated to its making working and other relevant details .

ELLIPTIC CALCULATOR:

- We have developed an application titled “Elliptic Calculator” whose basic functionality is to compute the various Cryptographic functionalities related to ECC .
- 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:

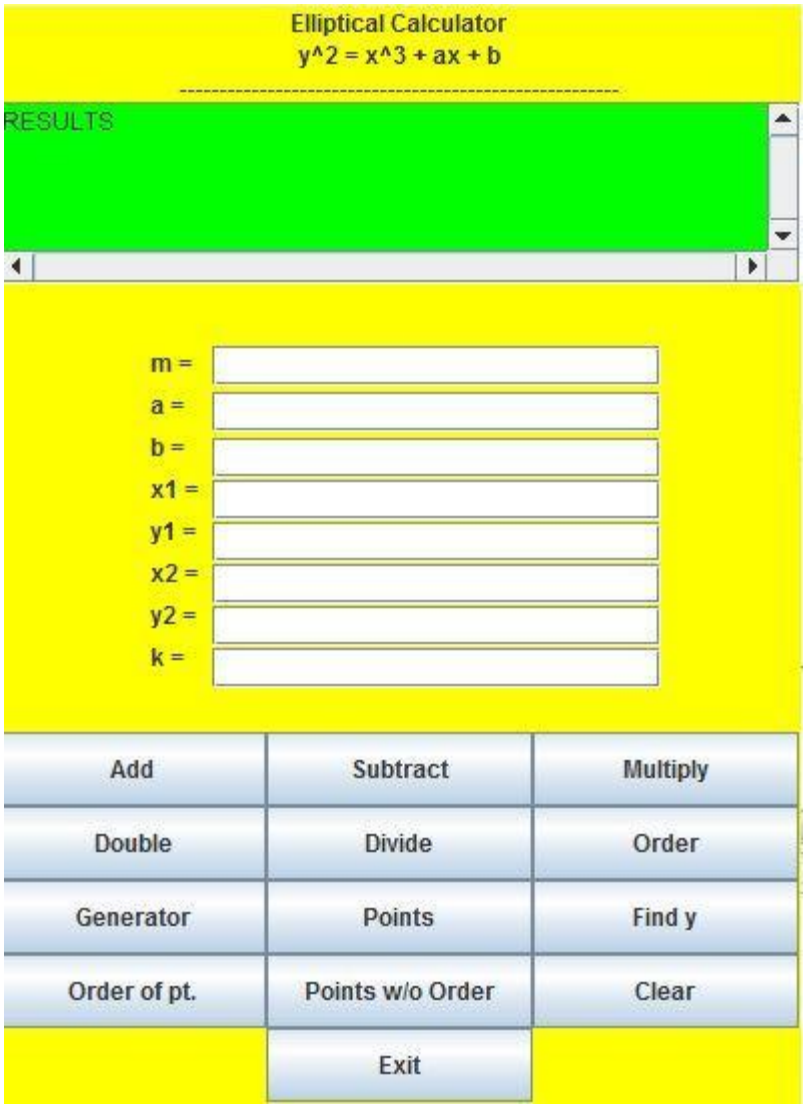
- This application can be used for the various ECC related calculations .

  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 8 text boxes for entering different values related to the elliptic curve.They are labelled as a,b,p,x1,y1,x2,y2 and k respectively . The notations used have their usual meanings .
- The results are displayed in a large text area which is uneditable.
- There are 13 command buttons for different operations labelled as add,sub,mull,double,Findy,Curveorder,genpoints,points,pts. w/oorder,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:

- Display Points without order
- Addition of 2 points

- 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

{

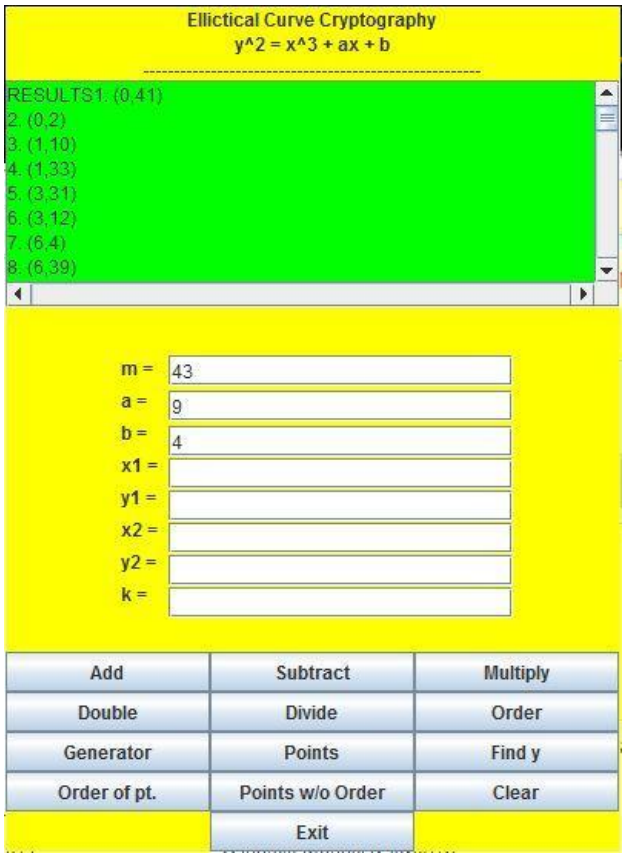
t1=p+1;t2=4;t1=t1/t2;y=xx mod p.

The generated pt is (x,y)

y=p-y . The generated pt is (x,y)

}

Screenshot of pts.w/o order :



The algorithm of Addition of 2 points :

Point addition is defined as taking two points along a curve E and computing where a line through them intersects the curve. We use the negative of the intersection point as the result of the addition.

The operation is denoted by:

$P + Q = R$ or,

$(x_P, y_P) + (x_Q, y_Q) = (x_R, y_R)$,

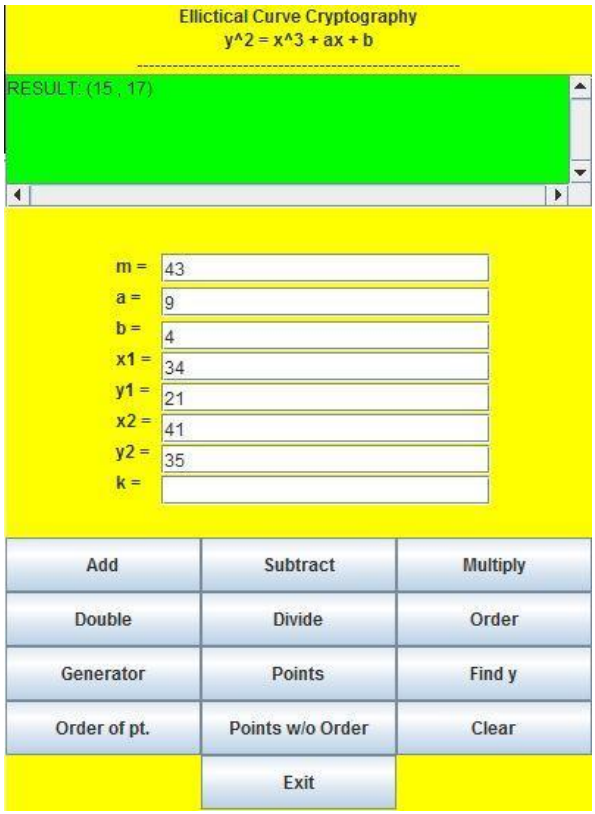
where

$x_R = L^2 - x_P - x_Q$ and,

$y_R = L(x_P - x_R) - y_P$ and,

$L = (y_P - y_Q)/(x_Q - x_P)$

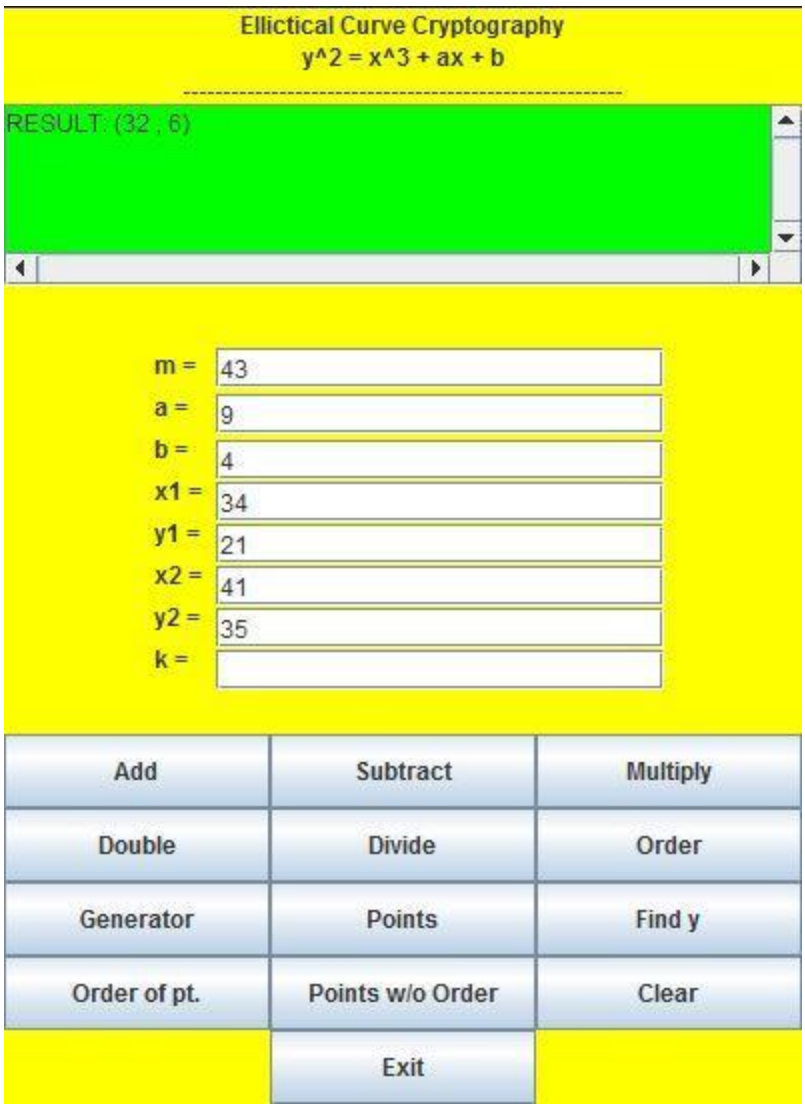
SCREENSHOT OF ADDITION :



Subtraction(The Algorithm) :

- 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)

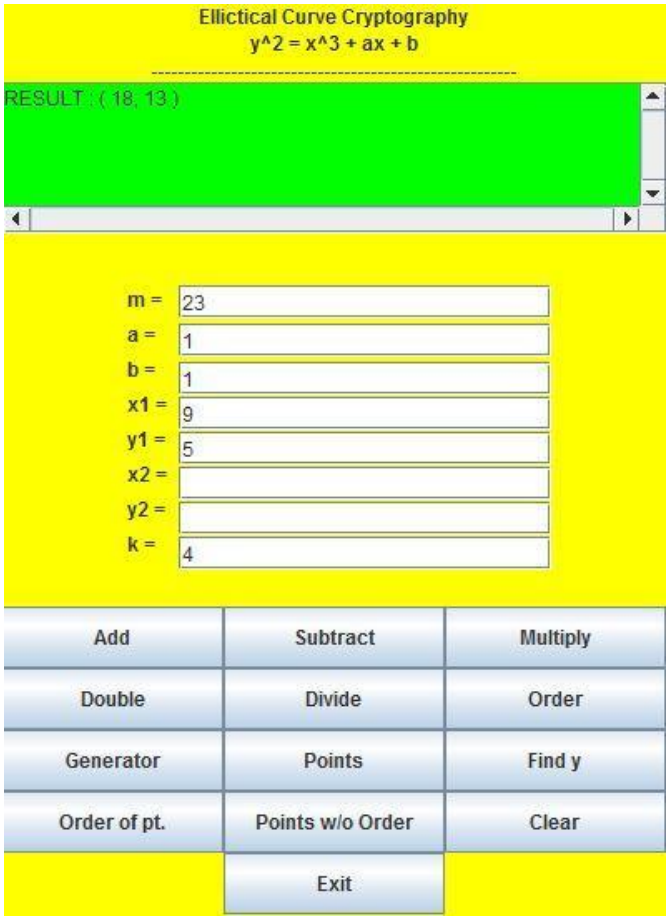
It can be expressed as (9,5) + (9,5) + (9,5) + (9,5).

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);

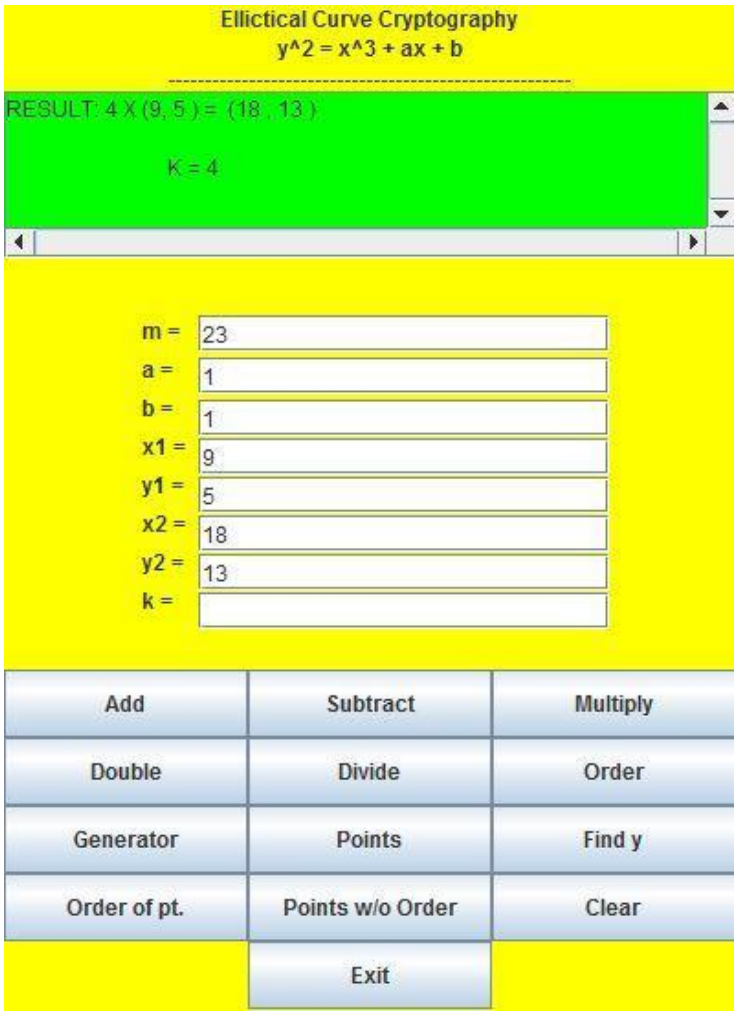
Screenshot of multiplication :



Division:

- **Division takes help of the Discrete logarithm problem.**
- **Consider the equation Q = (k * P),**
- **where Q, P ∈ 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:



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.

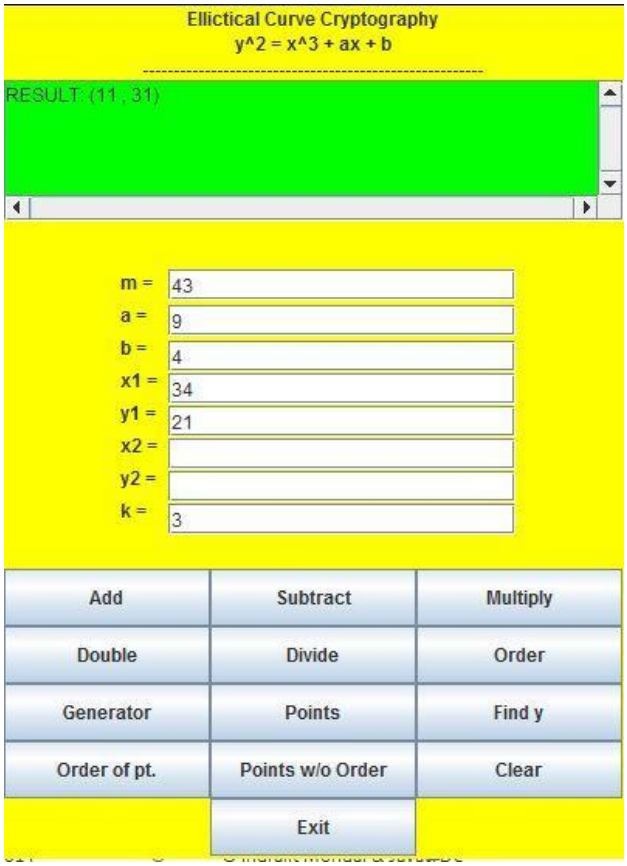
Geometrical explanation:

To double a point J to get L, i.e. to find L = 2J,

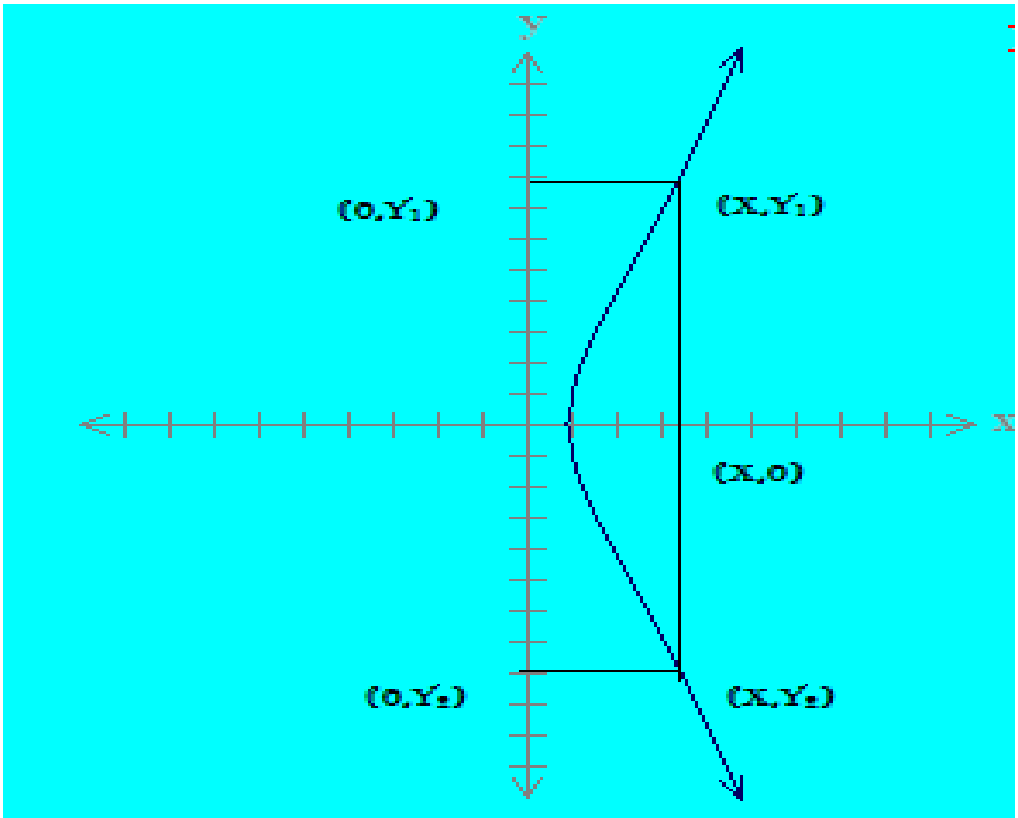
consider a point J on an elliptic curve as If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J.

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.

Screenshot of doubling :



Finding y coordinate for corresponding x coordinate :

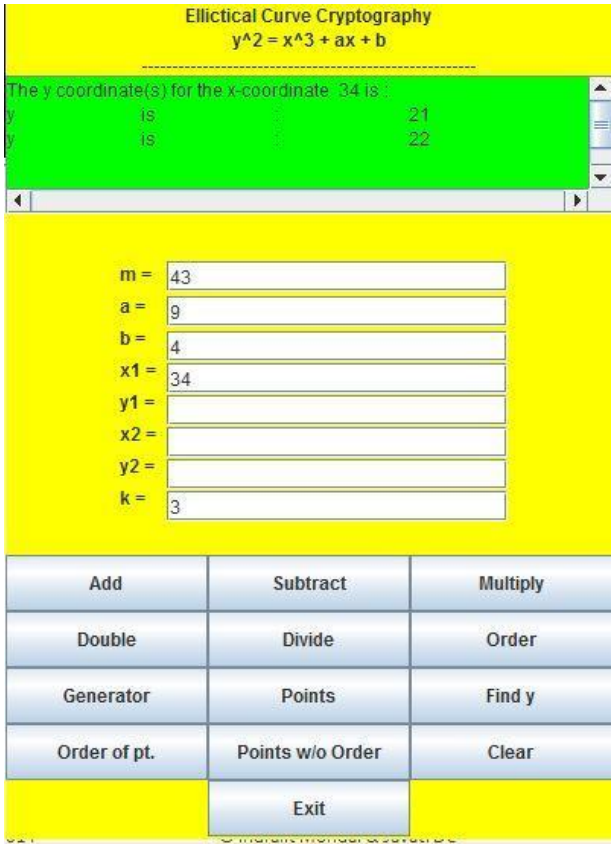


This method is all about

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) .

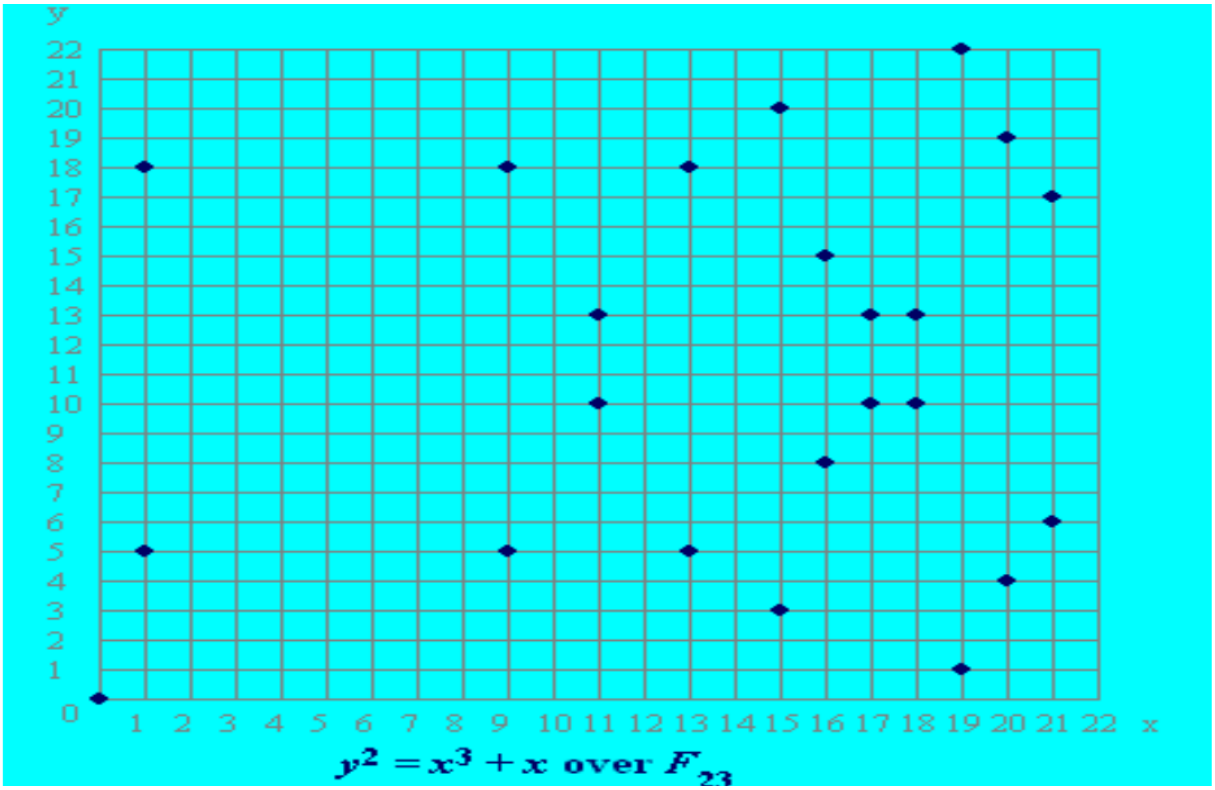
SCREENSHOT OF FIND Y :



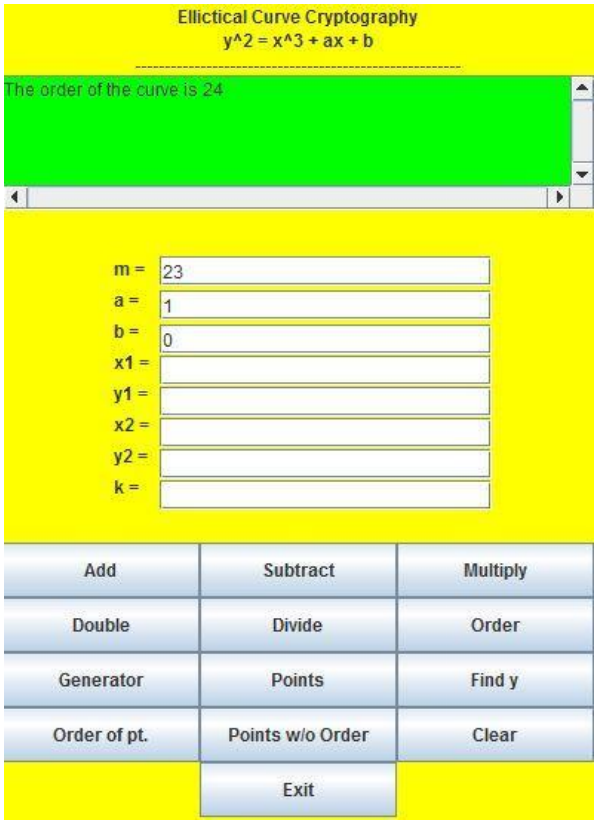
CURVE ORDER :

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:


y
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 x
$y^2 = x^3 + x$ over F_{23}

SCREENSHOT OF CURVE ORDER :



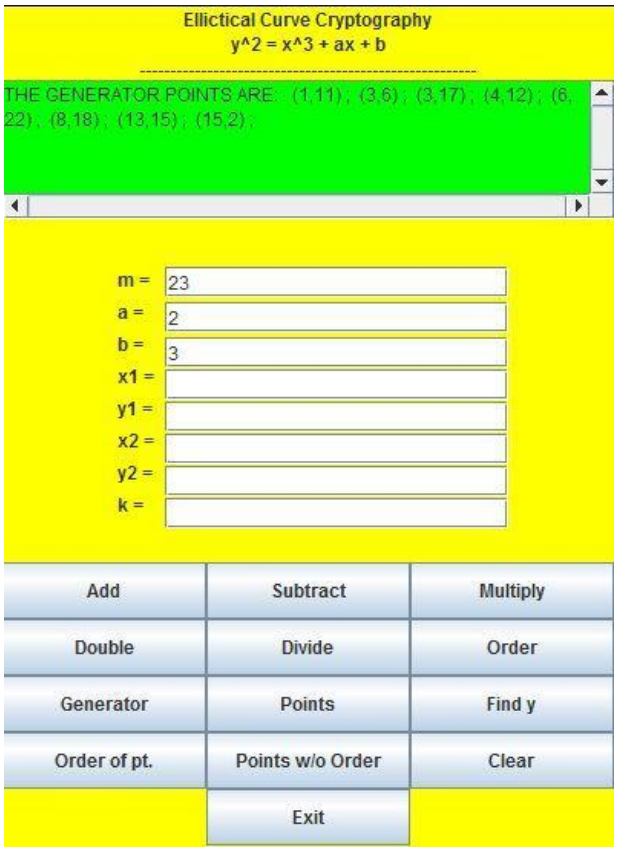
Gen Points (Generator points) :

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 .

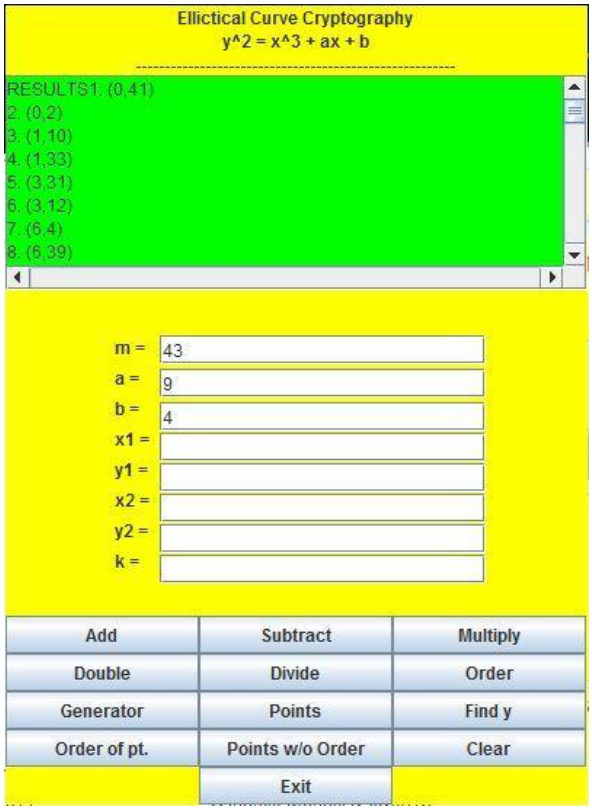
SCREENSHOT OF GENERATOR POINTS :



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

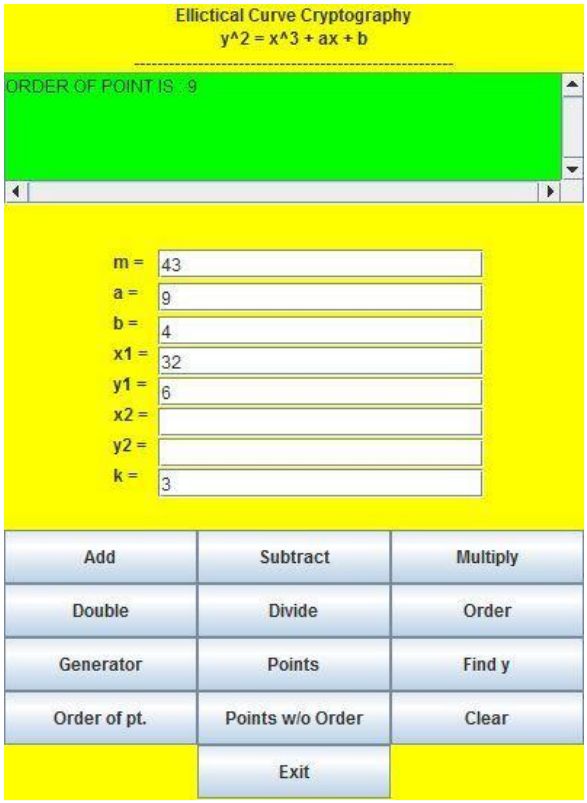
order=c;

break;

else

We add (x1,y1) with (xq,yq) & store the point in (xd,yd) }

SCREENSHOT OF ORDER OF A POINT



CONCLUSION

Thus we have finally developed an Elliptic curve based calculator which can perform many mathematical functions related to ECC .

Hence using this application we can verify whether any ECC based cryptosystem is working correctly or not . It can be used to generate values for Bruit force attack

used in Cryptanalysis .We have used Java for programming this application because of its platform independent nature . We have used the BigInteger class due to which our

application can handle large numbers . A jar file (and its corresponding Desktop Shortcut) have been created so that the application can be run without using netbeans.

this application can be run in both Windows and linux .

REFERENCES

- Lecture Notes & other materials supplied to us by our Mentor Prof. Utpal Kr. Ray.
- Cryptography & Network Security by B A Forouzan , D Mukhopadhyay
- Stallings,William, Cryptography and Network Security
- 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
- http://all.net/edu/curr/ip/Chap2-4.html