Elliptic Curve Cryptography

Presented by :-
Gaganjeet Reen(ICM2015003)

Problem Statement

Implement the following using your favorite programming language. Take a secure Elliptic Curve Ep(a, b), where p must be large (the term ‘large’ is quantified by your choice). Implement point addition and multiplication algorithms on this curve. Now, code an efficient algorithm which picks a randomly chosen point P from the curve and calculates and outputs the order of the point.

Do exhaustive experiments until you get a P such that Ord(P) = q, where q is a large prime.

Now, write another algorithm which will take a k-bit binary string ($k = \lceil \log2\ q \rceil$) as input, compute the decimal value d of the given string, return failure if the decimal value d is greater than (q-1), otherwise calculate Q = d.P .Experiment with the above algorithm to generate a large set of Q values (save all the outputs in a file) from randomly chosen k-bit strings.

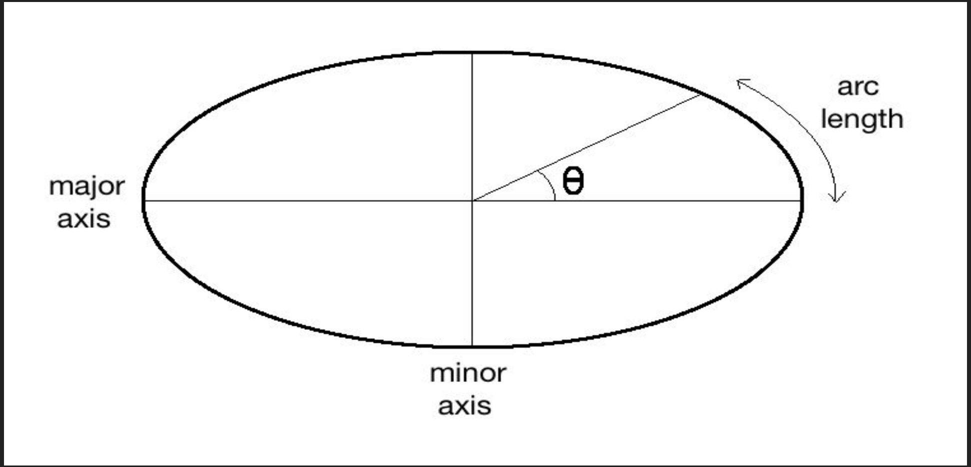
Problem Statement(Contd.)

Now, test the following:-

a) Whether the Q values are uniform-randomly distributed on the cyclic group generated by P.

b) Whether the Q values are uncorrelated even when there are high statistical correlations among the input k-bit strings.

Why are Elliptic Curves called Elliptic Curves ?



Given an ellipse. finding the arc length is complicated. Euler however, developed a function for it. It is a function involving integrals, square roots, and trigonometric functions. Mathematicians took this as a starting point and went on to describe an entire class of integral functions of similar form called Elliptic Integrals.

Contd.

Jacobi formulated a set of functions which can be used to invert Euler’s elliptic integrals i.e. given the length of the arc, find the angle. Jacobi’s functions will let us recover the angle of an elliptic arc given its arc length. These functions are called Elliptic Functions.

When we take the equations of elliptic functions defined over a particular space, we will find out that it has a limited set of possible solutions in that space. Now if we take these solutions (or points) and plot them, it forms a curve and it looks like a curve. These curves are called elliptic curves.

Elliptic Curves in Cryptography

Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields . Elliptic-curve cryptography requires smaller keys compared to non-EC cryptography to provide equivalent security.

The elliptic curve over Z_p , $p > 3$, is the set of all pairs $(x, y) \in Z_p$

which fulfill $y^2 \equiv x^3 + a*x + b \mod p$

together with an imaginary point of infinity O, where $a, b \in Z_p$

and the condition $4*a^3 + 27*b^2 \neq 0$ modp.

The definition of elliptic curve requires that the curve is nonsingular.

Algebraic Structures used in Elliptic Curve Cryptography

Two algebraic structures are used in Elliptic Curve Cryptography.

1. The field Z_p :- We define the elliptic curve on the field Z_p instead of Real Numbers. p here is defined to be a large prime number.
2. A group with the set of points as the points on elliptic curve in Z_p and the group operation as Point Addition. Point addition is designed in a way such that it satisfies all the group properties like :- Closure, Existence of Inverse, Existence of Identity and Associativity.

Choosing Secure Elliptic Curves

There are a few basic criteria that "secure" curves must fulfill. Matching all these criteria does not guarantee security in any way, but if any of them is not matched, then the curve is definitely not secure.

- The curve order (number of points on the curve) must be a large enough prime integer, or a multiple of a large enough prime integer. "Large enough" means here that the Elliptic Curve Discrete Logarithm problem should not be easily solvable by brute force. This is required because according to Cauchys theorem, if *G* is a finite group and *q* is a prime number dividing the order of *G* then *G* contains an element of order *q*, and the order- *q* element generates an order *q* subgroup..

Contd.

- The curve should not be singular. This basically means that there should be no point of intersection on the elliptic curve.

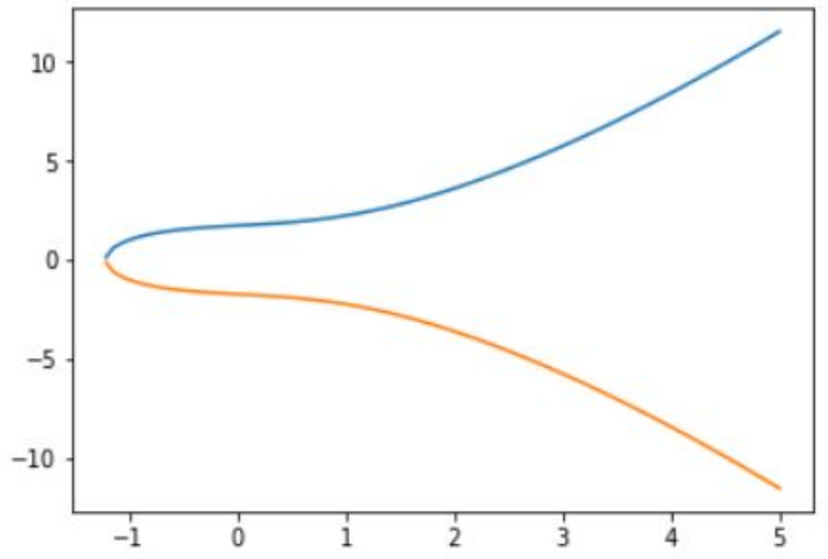


Fig. 3. Elliptic Curve over the field of real numbers with the chosen parameters i.e a=1, b=3, p=15733

Singular vs Non-Singular Elliptic Curves

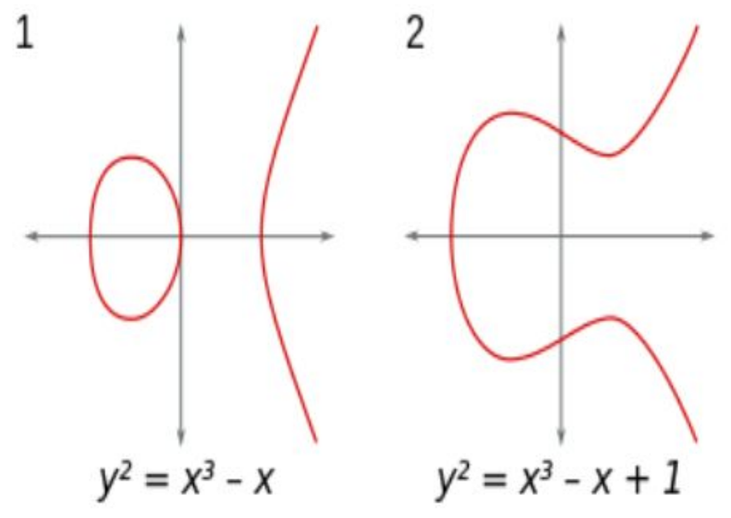


Fig. 1. Non singular elliptic curve

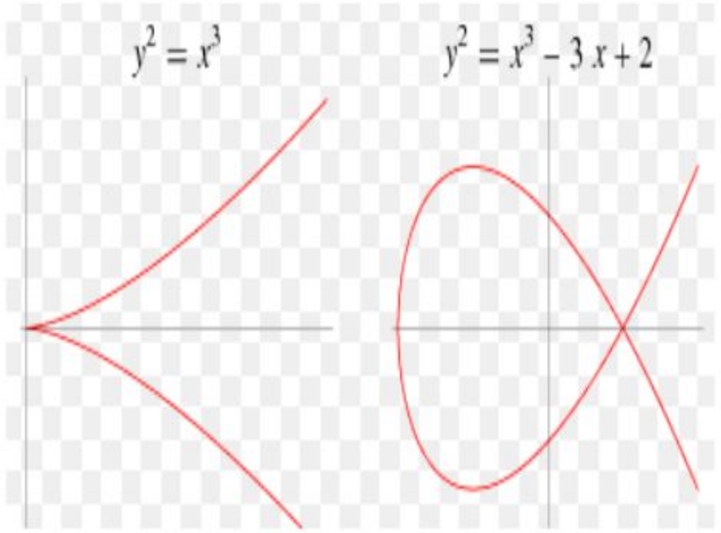
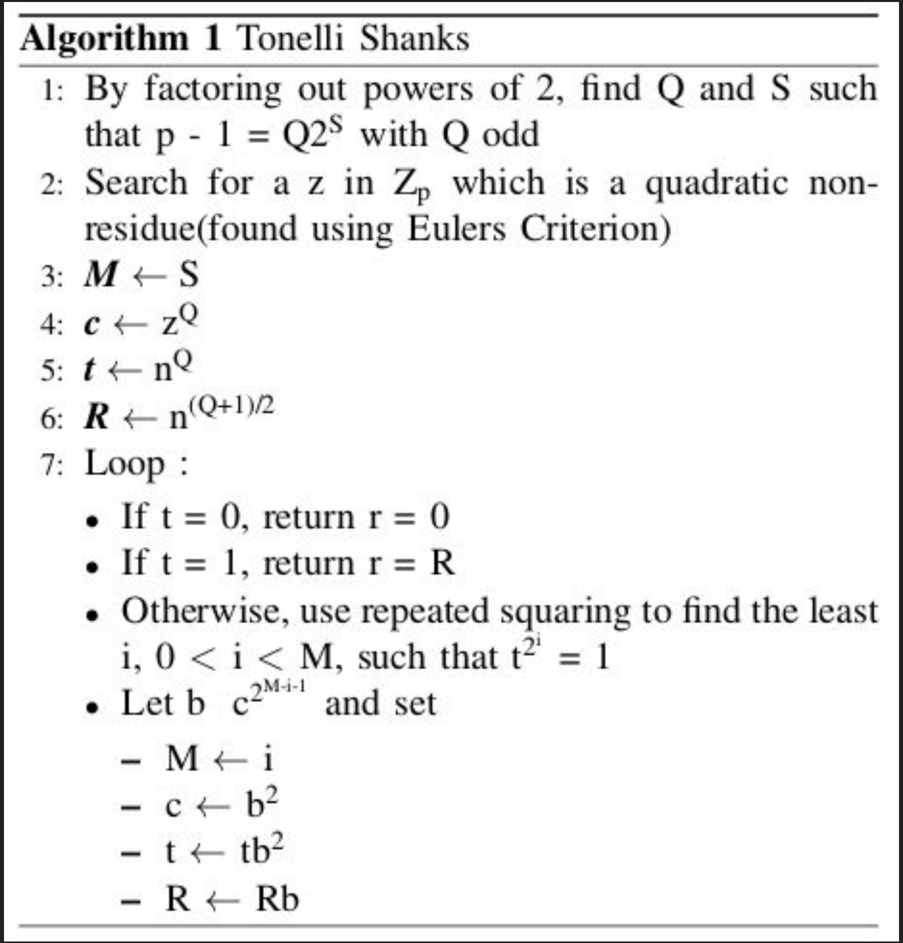


Fig. 2. Singular elliptic curve

Computing points on an Elliptic curve over Z_p



Algorithm 1 Tonelli Shanks

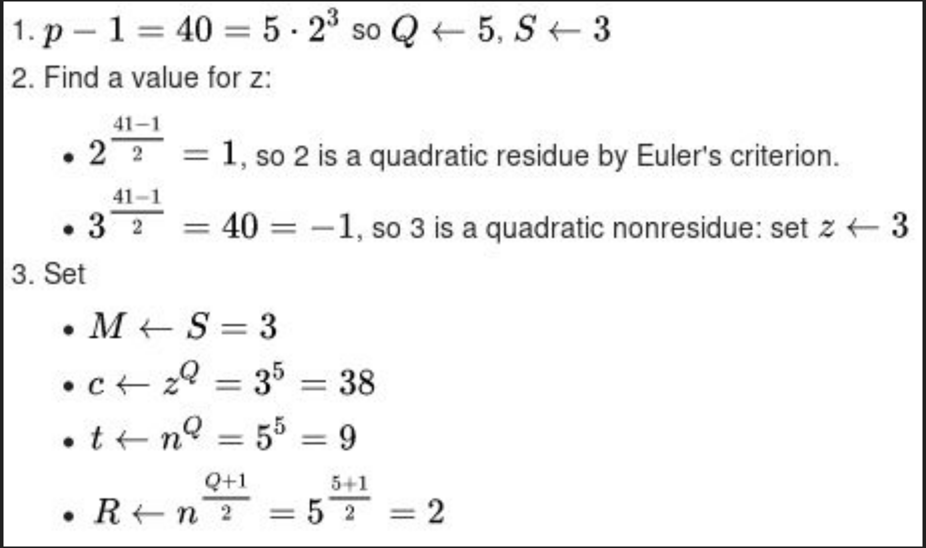
```
1: By factoring out powers of 2, find Q and S such
   that p - 1 = $Q2^S$ with Q odd
2: Search for a z in $Z_p$ which is a quadratic non-
   residue(found using Eulers Criterion)
3: $\boldsymbol{M} \leftarrow S$
4: $\boldsymbol{c} \leftarrow z^Q$
5: $\boldsymbol{t} \leftarrow n^Q$
6: $\boldsymbol{R} \leftarrow n^{(Q+1)/2}$
7: Loop :
     • If t = 0, return r = 0
     • If t = 1, return r = R
     • Otherwise, use repeated squaring to find the least
       i, $0 < i < M$, such that $t^{2^i} = 1$
     • Let b  $c^{2^{M-i-1}}$ and set
         – $M \leftarrow i$
         – $c \leftarrow b^2$
         – $t \leftarrow tb^2$
         – $R \leftarrow Rb$
```

Finding the points that lie on an elliptic curve over Z_p requires calculation of square root under modulo p. Tonelli Shanks algorithm can be used to do this as long as p is an odd prime(a major reason for choosing p > 3).

Example of how the Tonelli Shanks Algorithm works :-

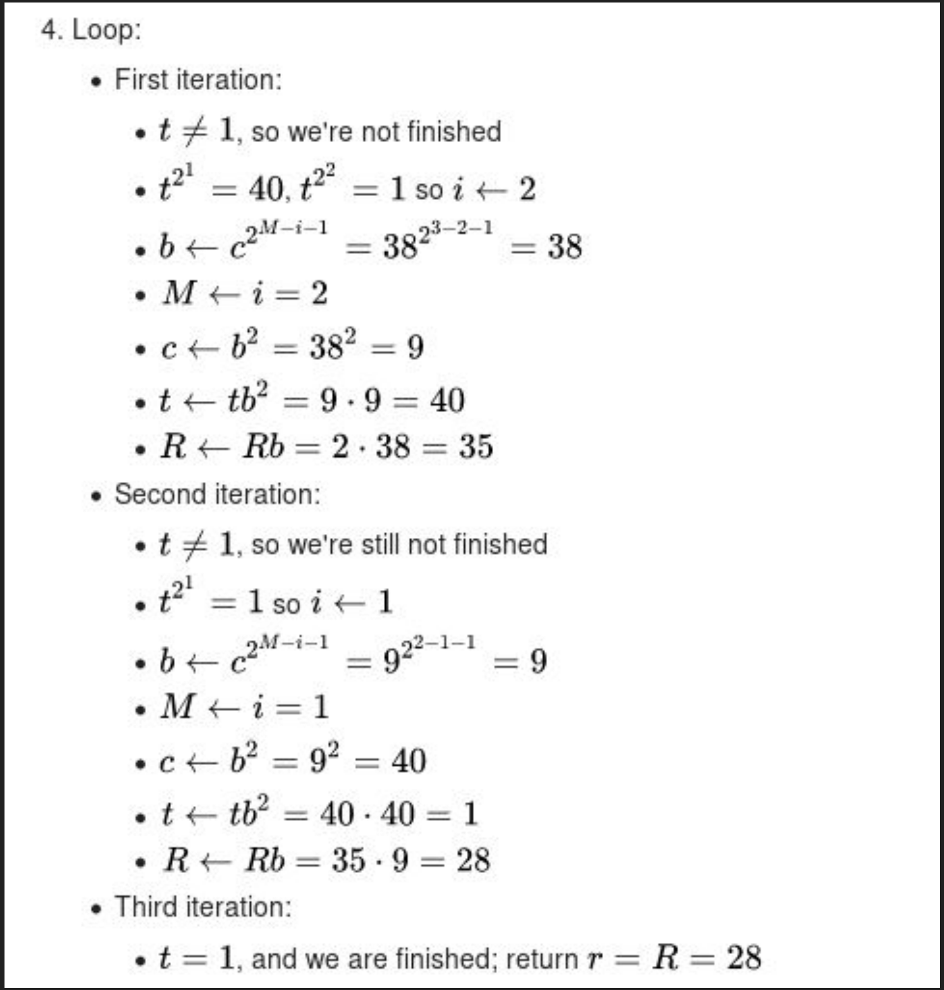
Solving the congruence $r^2 \equiv 5 \pmod{41}$.

41 is prime as required and $41 \equiv 1 \pmod 4$. 5 is a quadratic residue by Euler's criterion: $5^{41-1/2} = 5^{20} = 1 \bmod 41$. All the following operations are performed under mod 41.



1. $p - 1 = 40 = 5 \cdot 2^3$ so $Q \leftarrow 5, S \leftarrow 3$
2. Find a value for z:
   - $2^{\frac{41-1}{2}} = 1$, so 2 is a quadratic residue by Euler's criterion.
   - $3^{\frac{41-1}{2}} = 40 = -1$, so 3 is a quadratic nonresidue: set $z \leftarrow 3$
3. Set
   - $M \leftarrow S = 3$
   - $c \leftarrow z^Q = 3^5 = 38$
   - $t \leftarrow n^Q = 5^5 = 9$
   - $R \leftarrow n^{\frac{Q+1}{2}} = 5^{\frac{5+1}{2}} = 2$

Contd.



4. Loop:
   - First iteration:
     - $t \neq 1$, so we're not finished
     - $t^{2^1} = 40$, $t^{2^2} = 1$ so $i \leftarrow 2$
     - $b \leftarrow c^{2^{M-i-1}} = 38^{2^{3-2-1}} = 38$
     - $M \leftarrow i = 2$
     - $c \leftarrow b^2 = 38^2 = 9$
     - $t \leftarrow tb^2 = 9 \cdot 9 = 40$
     - $R \leftarrow Rb = 2 \cdot 38 = 35$
   - Second iteration:
     - $t \neq 1$, so we're still not finished
     - $t^{2^1} = 1$ so $i \leftarrow 1$
     - $b \leftarrow c^{2^{M-i-1}} = 9^{2^{2-1-1}} = 9$
     - $M \leftarrow i = 1$
     - $c \leftarrow b^2 = 9^2 = 40$
     - $t \leftarrow tb^2 = 40 \cdot 40 = 1$
     - $R \leftarrow Rb = 35 \cdot 9 = 28$
   - Third iteration:
     - $t = 1$, and we are finished; return $r = R = 28$

We can confirm that, $28^2 \equiv 5 \pmod{41}$ and $(-28)^2 \equiv 13^2 \equiv 5 \pmod{41}$. So the algorithm yields the two solutions to our congruence i.e. 28 and -13.

Point Addition and Point Multiplication(Doubling)

Given two points and their coordinates, say P = (x1 , y1) and Q = (x2 , y2), we have to compute the coordinates of a third point R such that: P + Q == R

(x1 , y1) + (x2, y2) == (x3 , y3)

Analytically :-

$x3 = s^2 - x1 - x2 \bmod p$ ..(1)

$y3 = s(x1 - x3) - y1 \bmod p$ ..(2)

where $s = (y2 - y1 / x2 - x1) \bmod p$; if $P \neq Q$ (point addition) ..(3)

and $s = ((3x1^2 + a) / 2y1) \bmod p$; if $P = Q$ (point doubling) ..(4)

Derivations of Point Addition Formulae

Let P(x1,y1),Q(x2,y2) and S(x4,y4) be three points on a straight line intersecting the curve at points P , Q and S.

The equation of the line through P and Q will be :-

$y = y1+s(x-x1)$ ..(5)

Substituting this into $y^2 = x^3 + a*x + b$ ..(6)

we get,

$x^3 - s^2x^2 + (a+2*s^2*x1-2*s*y1)x+b-(s*x1-y1)^2 = 0$ ..(7)

Contd.

The three solutions to that cubic equation give the x-coordinates $x1,x2,x4$ of the three points of intersection of the line with the curve.From Vieta's first formula, we see that the sum of those x-coordinates is $s2$ so that $x1+x2+x4=s2$. When we reflect S over the x-axis, the x-coordinate does not change, so $x3=x4$. Thus,

$$x3=s2-x1-x2. \quad ..(8)$$

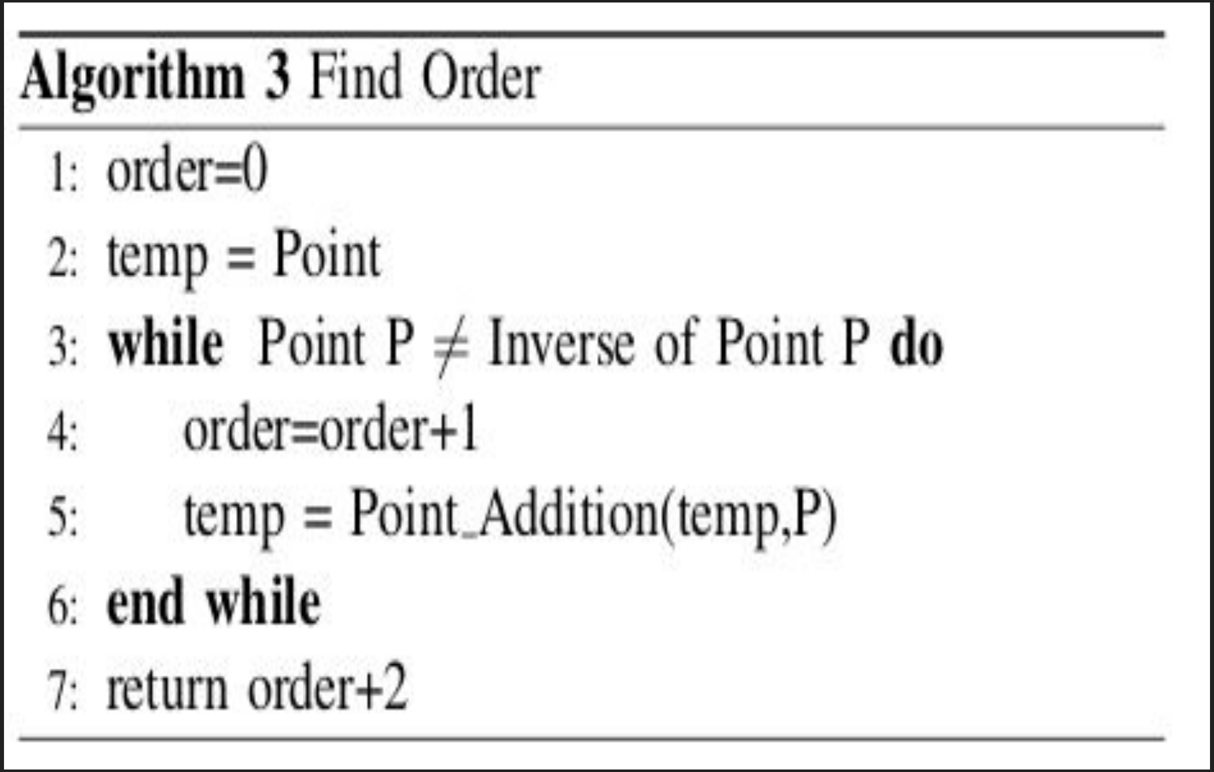
Using the equation of the line, $y4=y1+s(x4-x1) \quad ..(9)$

$$=y1+s(x3-x1) \quad ..(10)$$

When we reflect S over the x-axis, the sign of the y-coordinate changes, i.e., $y3=-y4$. Thus,

$$y3=s(x1-x3)-y1. \quad ..(11)$$

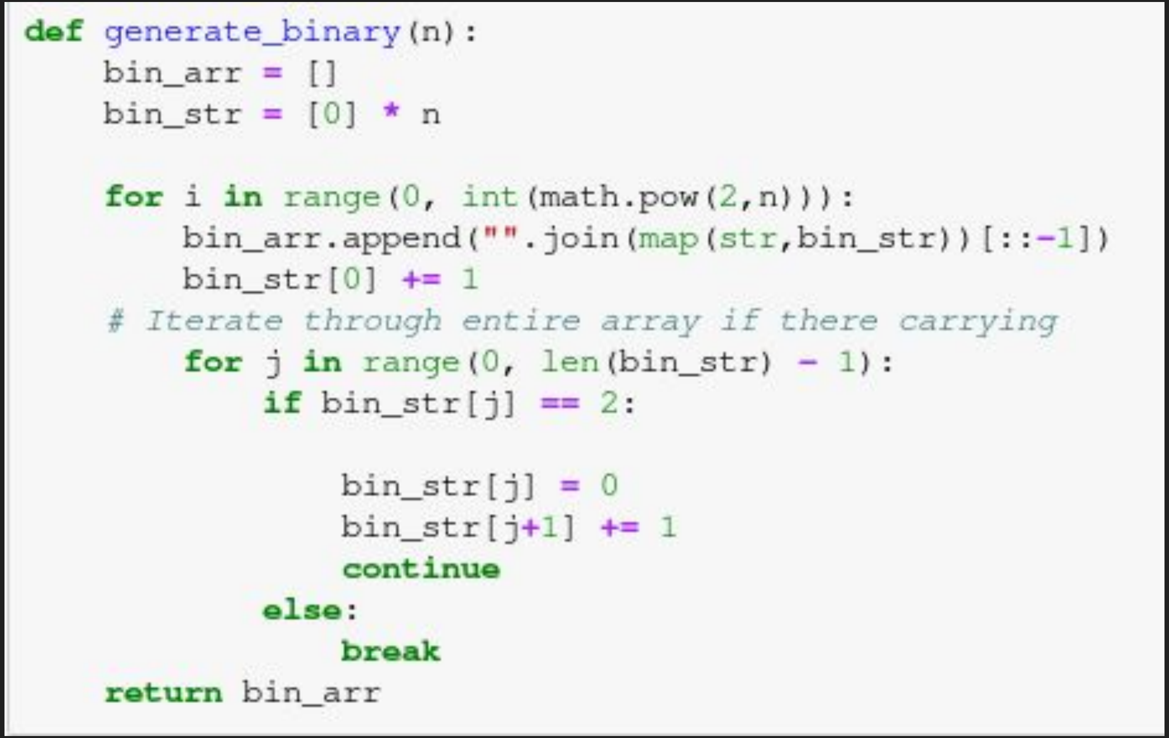
Finding the order of a point on the curve



Algorithm 3 Find Order

```
1: order=0
2: temp = Point
3: while  Point P $\neq$ Inverse of Point P do
4:     order=order+1
5:     temp = Point_Addition(temp,P)
6: end while
7: return order+2
```

Generating k-bit binary strings



```python
def generate_binary(n):
    bin_arr = []
    bin_str = [0] * n

    for i in range(0, int(math.pow(2,n))):
        bin_arr.append("".join(map(str,bin_str))[::-1])
        bin_str[0] += 1
    # Iterate through entire array if there carrying
        for j in range(0, len(bin_str) - 1):
            if bin_str[j] == 2:

                bin_str[j] = 0
                bin_str[j+1] += 1
                continue
            else:
                break
    return bin_arr
```

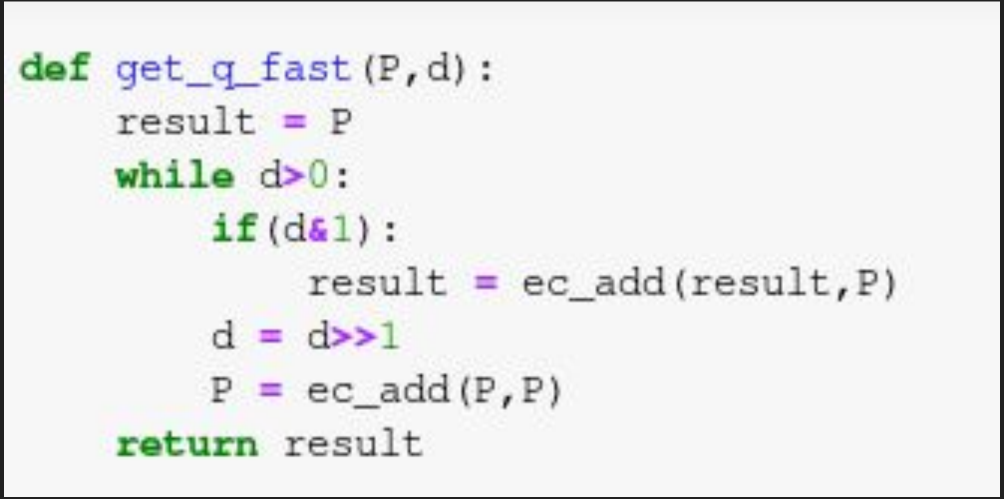
Selecting 5000 random strings from the generated k-bit strings

The secrets module of python is used to generate 5000 random numbers lesser than the order of the selected point(P) and corresponding to these random numbers, the points are generated on the elliptic curve using Q = d.P and stored in a file.

The secrets module is used because it generates cryptographically secure random numbers.The secrets module is based on os.urandom() and random.SystemRandom() functions, which are the interface to the operating system’s best source of cryptographic randomness. However, it should be noted that this module does not generate deterministic random numbers, This means that the sequence generated can not be repeated using some seed value.

Generating the points(Q)

The square and multiply algorithm is used to generate the points on an Elliptic Curve.



```python
def get_q_fast(P,d):
    result = P
    while d>0:
        if(d&1):
            result = ec_add(result,P)
        d = d>>1
        P = ec_add(P,P)
    return result
```

Testing whether the points are uniform-randomly distributed over the cyclic group generated by P

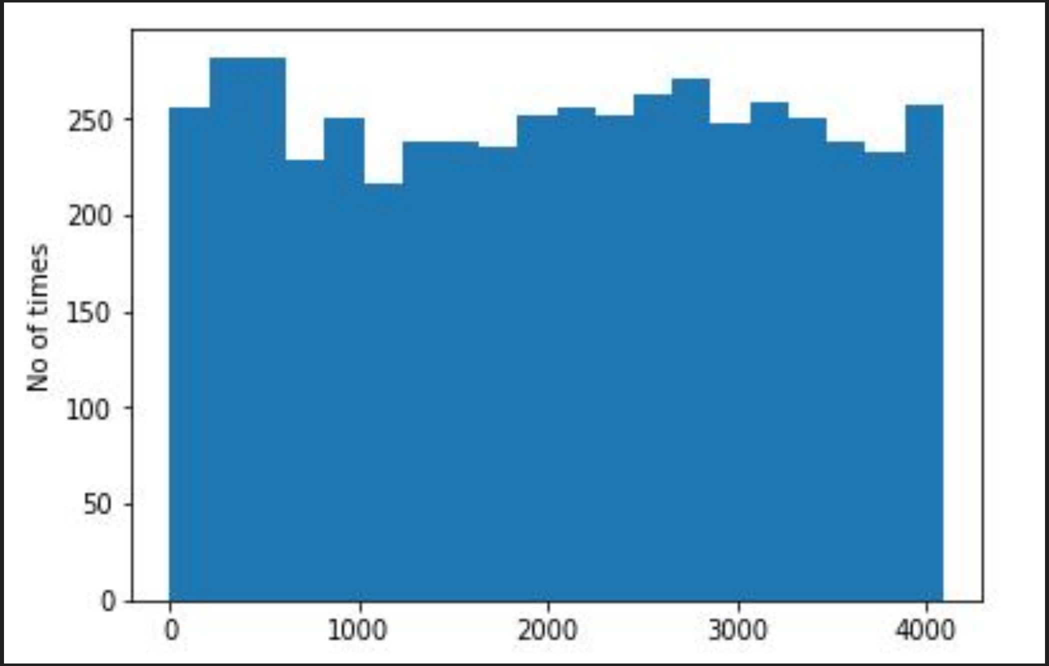
The following methods have been used to test the uniform-randomness of points :-

1. Histogram Analysis
2. Chi Square test for independence
3. Voronoi Diagrams to test randomness

Since all the randomly generated strings have decimal values are lesser than the order of P, if we can prove that they have a uniform random distribution, we can also say that the points generated will have the same distribution as the random numbers generated above. This is because each decimal value(lesser than the order of the cyclic group) maps to a unique point in the cyclic group generated by P.

Histogram Analysis

When we create a histogram for the generated random numbers, it looks like :-



From the histogram we can conclude that the generated numbers follow a roughly uniform distribution. The bin size for the given histogram was set to 20. The fact that the distribution is not perfectly uniform assists in proving the randomness of the numbers as well.

Chi Square test for Independence

In order to perform the chi square test for independence, we first prepare a dictionary of the generated numbers with the number as the key and the frequency of the numbers as the value. Then we test for independence between the numbers by calculating the p-value and comparing it with the value of the significance level(0.05).

Contd.

p-value $\leq \alpha$ implies that the variables have a statistically significant association (Reject H_0)

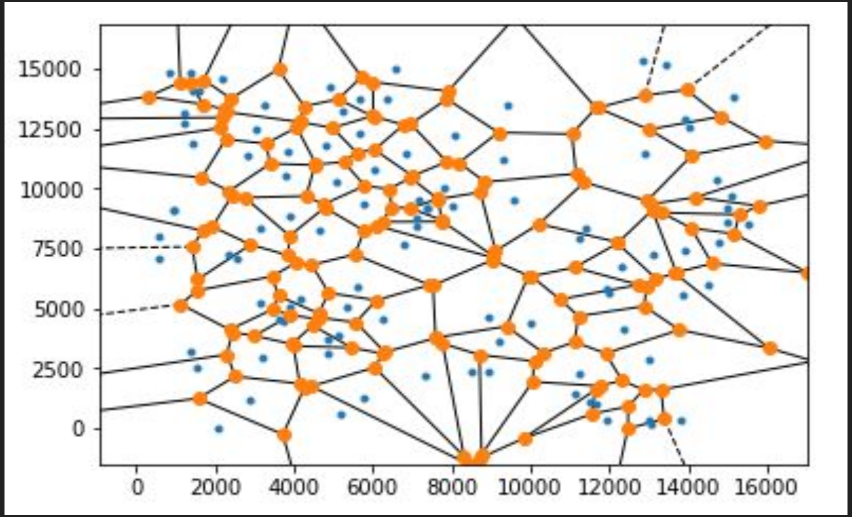
p-value $> \alpha$ implies that we cannot conclude that the variables are associated (Fail to reject H_0)

Here H_0 is the null hypothesis which states that the variables are independent of each other.

Since in our case p-value evaluates to be 0.60488 which is greater than 0.05, we fail to reject H_0

Voronoi Diagrams

The randomness can also be observed in terms of the Voronoi Diagrams. The Voronoi diagram partitions R^2 into regions based on the samples. Each sample x has an associated Voronoi region Vor(x). For any point $y \in \text{Vor}(x)$, x is the closest sample to y using Euclidean distance. The different sizes and shapes of these regions in the figure below gives some indication of the randomness of the points obtained.



Demonstrating that Q values are uncorrelated even when there are high statistical correlations among the input k-bit strings.

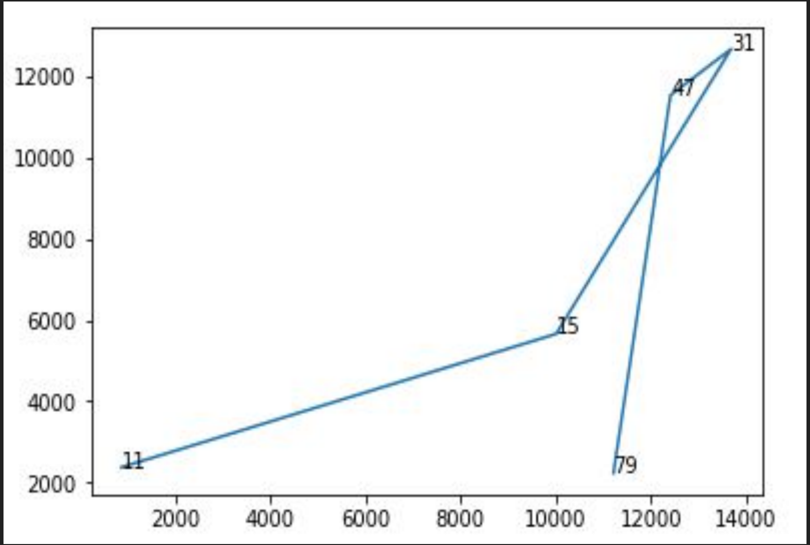
We have made use of Pearsons Correlation Coefficient to demonstrate the above mentioned statement.
The value of the coefficient ranges from -1 to 1 with -1 indicating high negative correlation. +1 indicating high positive correlation and 0 indicating no correlation.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,]\,[\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

Here, r is the pearsons correlation coefficient and x and y represent the variables between which we are trying to calculate the correlation.

Contd.

By trial and error, we first pick a set of strings such that all of them have a high correlation between themselves of greater than 0.5. These decimal values of the selected strings are:- 11,15,31,47 and 79 respectively.



From the plot it is evident that there is no predictable pattern in the points plotted even though the strings whose decimal values generated these points have a high statistical correlation.

The correlation values between the strings and the x-coordinates of the points is -0.037998 and

the strings and y-coordinates is -0.04553.

Thus we can conclude that there is no correlation between the points and the decimal values that generate them.

Contd.

We have also made use of Spearmans Correlation Coefficient to calculate the correlations.
The value of the coefficient ranges from -1 to 1 with -1 indicating high negative correlation. +1 indicating high positive correlation and 0 indicating no correlation.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Here, ρ is the Spearmans correlation coefficient,
where
d = the pairwise distances of the ranks of the variables x_i and y_i and
n = the number of samples

Contd.

We calculate the coefficient for x coordinates and the y coordinates of the points separately. The values of Spearman coefficients are :-

- -0.03715
- -0.04596

for x coordinates and y coordinates respectively. These values show that there is almost no correlation between the decimal values and the points obtained on the Elliptic Curve under modulo p corresponding to these decimal values.

Conclusion

From the above experiments, we can conclude the following :-

- If we plot an elliptic curve over Z_p , we do not get anything remotely resembling a curve. For an Elliptic Curve to be cryptographically secure, the ECDLP should be hard for a given elliptic curve. For the ECDLP problem to be hard, we need a large enough Cyclic Group on the Elliptic Curve over Z_p. The security of the curve largely depends on the values of the parameters a, and b, the value of p(which should ideally be a large prime) and on the number of points on an elliptic curve over the field Z_p. The number of points should ideally be a large prime number so as to avoid a small subgroup attack.

Contd.

- The fact that there is almost no correlation between the values of d and Q in Q=d.P, also ensures the difficulty of the ECDLP. Thus, given Q and P, it is impossible to determine d other than by using the brute force method. This characteristic of Elliptic Curves facilitates their use in Public Key Cryptography where Q is usually the public key and d is the private key.