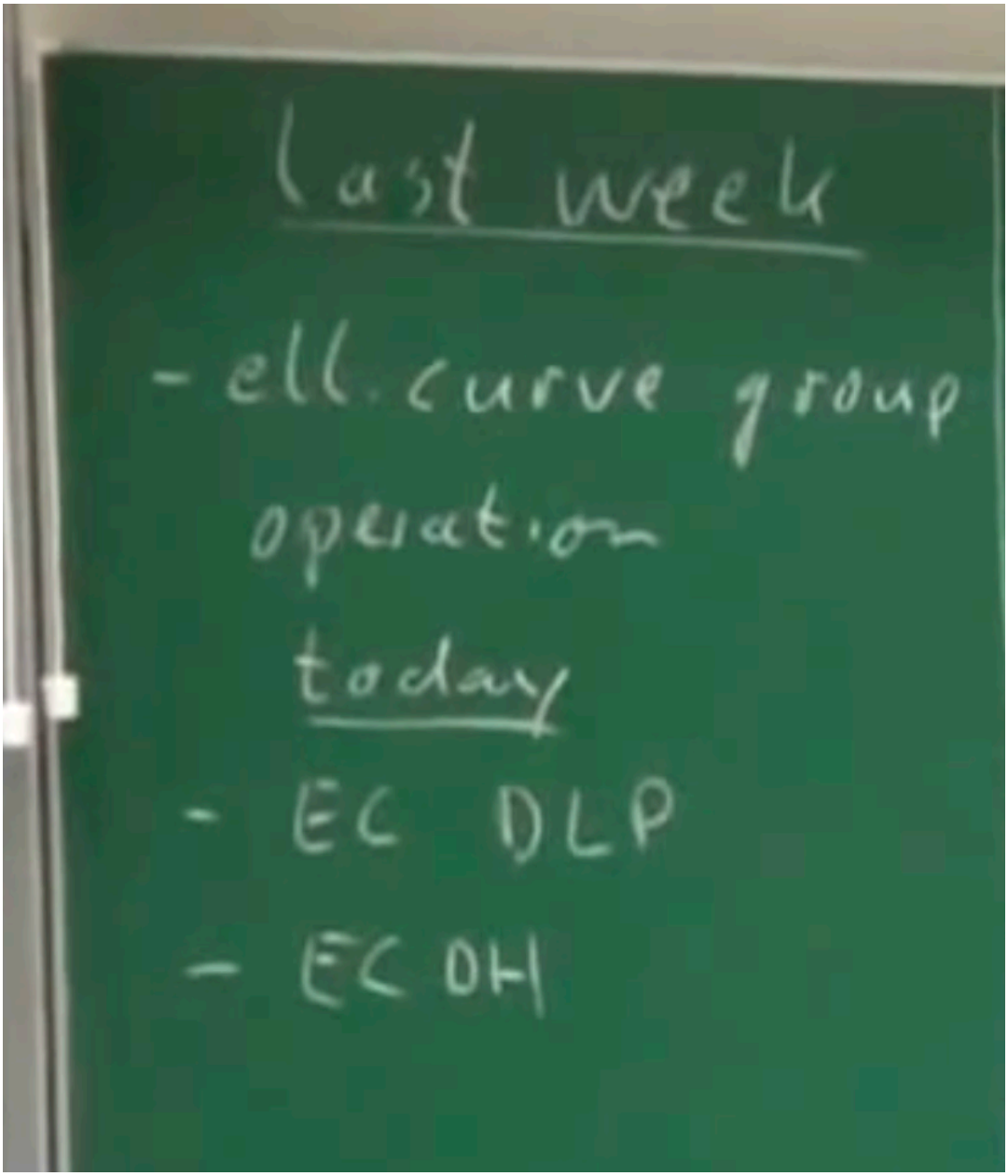
Lecture 17. Elliptic Curve Cryptography

Last Week we had learned about Elliptic curve group operations.

Today:

1. EC DLP -> Elliptic curve discrete logarithm problem which is quite interesting one way function.
2. EC DH -> Elliptic curve diffie hellman



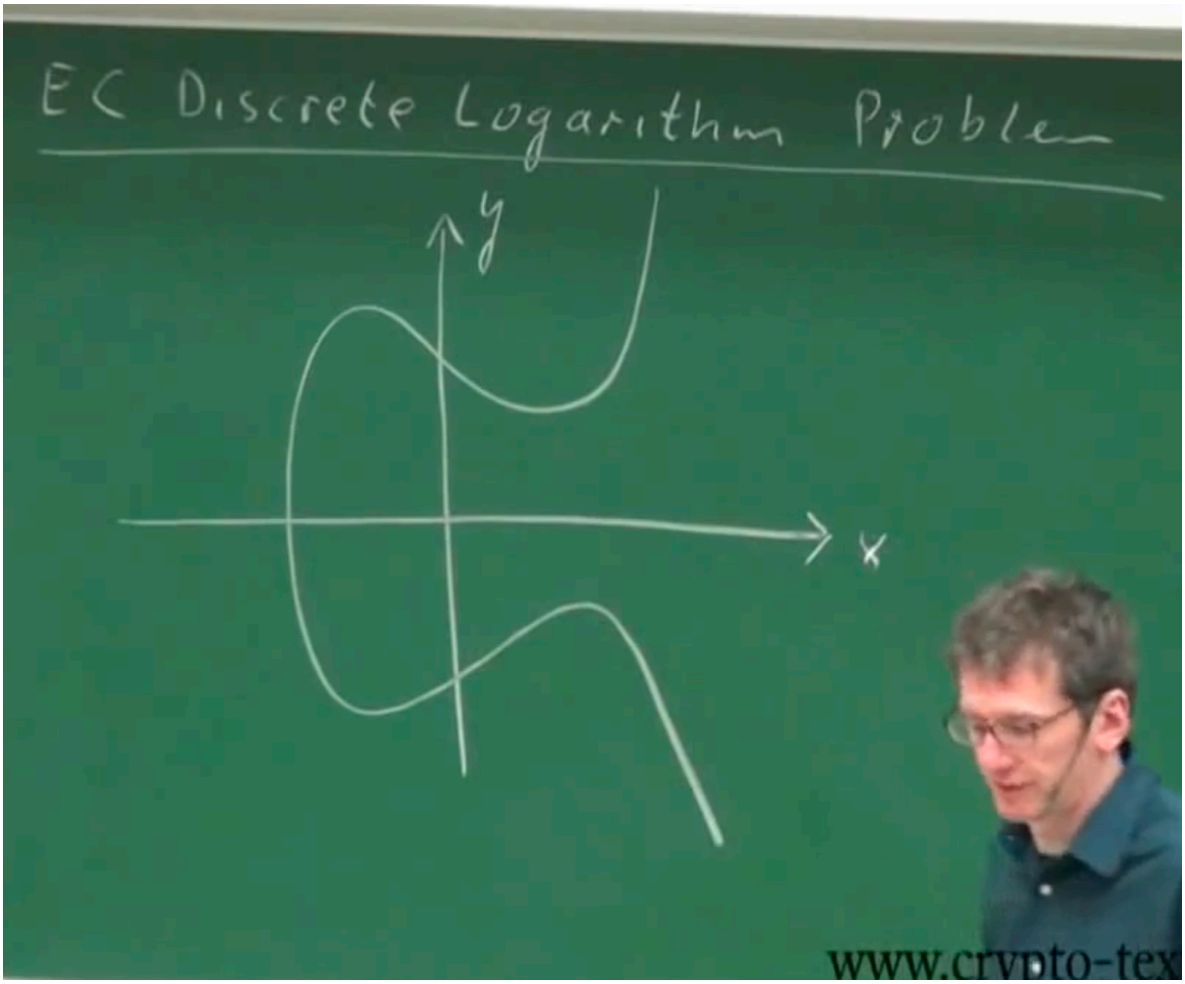
One Question i have is can we use DLP as a HashFunction as this is also a one way function ?

We have already studied DH and DLP in last 4 lectures so in this lecture and we will try to use these using EC ie different structures. we combine this weird EC with DLP.

Also with Diffie Hellman, we take this complicated stuff ie **EC with DLP** and related it to Diffie Hellman to get the better understanding on what is DLP and what is Diffie Hellman.

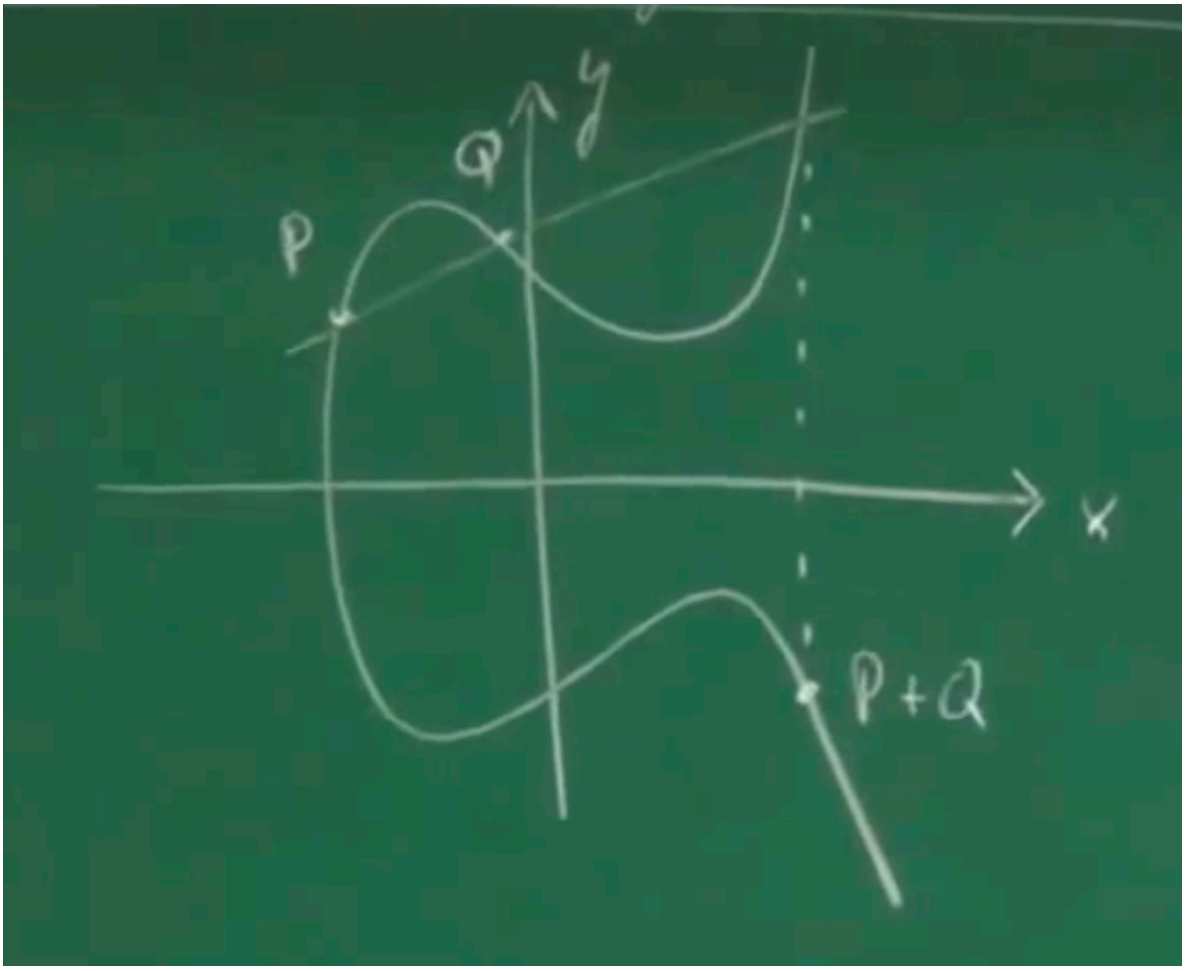
Chapter 1: EC Discrete Logarithm Problem

In last lecture we have shown that if we consider some kind of equation over real numbers we get following type of curve:



then consider there are two points P and Q, we found P+Q ie addition of Points in EC.

eg:



So last week we talked that they form the cyclic group under certain conditions. So let's think more on it.

E.g. EC as a cyclic group
E : Y^2 = X^3 + 2*X + 2 mod 17

For this specific curve all the points on the curve form a cyclic group which is not always true.

Question: What is the requirement for cyclic group ie what kind of thing do we need to have for cyclic group ?
Generator or Primitive Element.

For our equation we are taking Primitive element P = (5,1) -> **Important point is it is a "point"**

Question what is the primitive element ?
in case of Multiplicative group if we keep on taking power/multiple we generate entire group.

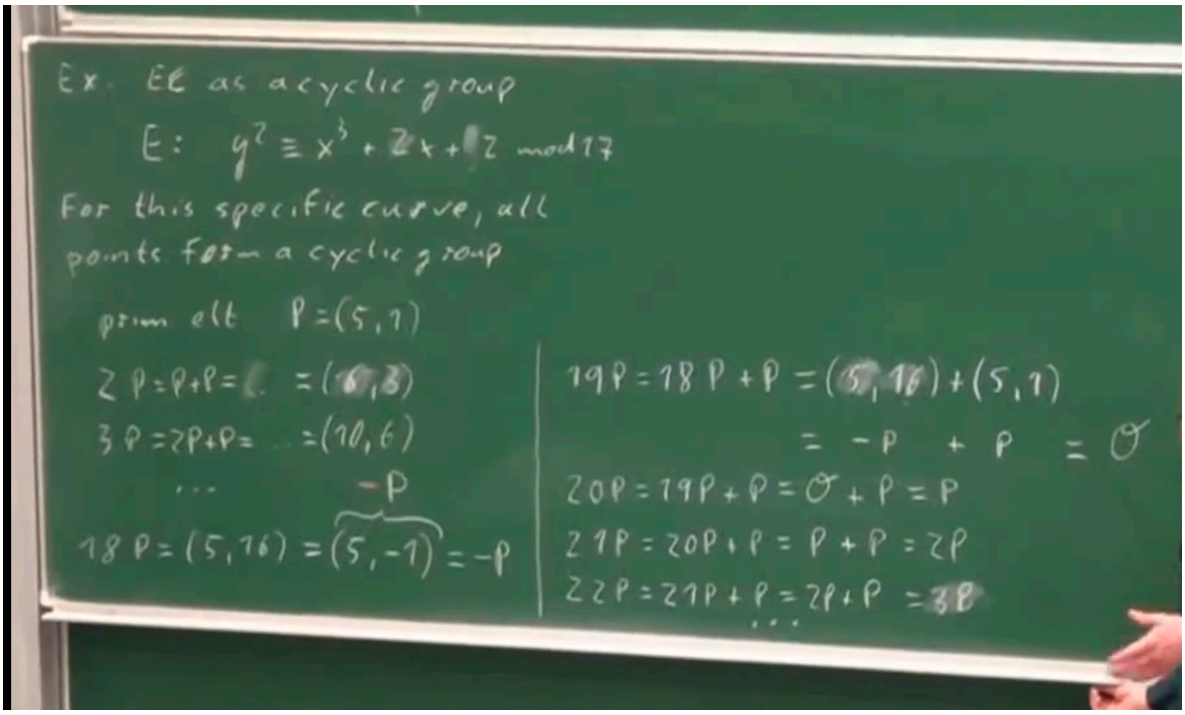
ie a, a*a, a*a*a, ... => a, a^2, a^3 ...

so in case of additive group it will be a, 2*a, 3*a ...

so 2P = P + P = (6,3)
3P = 2P + P = (10,6)
....
18P = (5,16) in this X coordinates are same this means this line is parallel to Y axis ... **Important Observation.**

Question, How are the 16 and 1 related ?
Answer they are the inverse, Modulo 17 as 16 mod 17 is same as -1 mod 17

so we can write
18P = (5, -1) which is **-P**
19P = P + -P = Infinity
20P = Infinity + P = P so we reach the starting point which is proving that it is a cyclic group.



So if you see we are gain back to P and we are cycling here and we have a cyclic group.

Important point here is that, This cyclic group is so different where we are doing it in a group of Points and when we looked at cyclic group in Zp* we might have thought that it is a weird behaviour which is there with numbers and modulus but here we are getting the same behaviour with points.
so when we generalise things we get deeper insight.

We are not that fascinating with groups with cyclic structure but we want to do something with that ie we want to build cryptosystems.

for building cryptosystem we require certain kind of problems so in this case we are building Discrete Logarithm problem.

Finding DLP is quite easy as we have already looked at DLP in Zp*.
So Lets related this to DLP in Zp*
if say group operation in Zp* DLP is "." and generator is "a" then
DLP will be **a.a.a.a.a.a..... some x times mod p = Kpub** so if we make "." operation as Multiply then it becomes DLP for Zp* .
Now in EC lets make "." as additive so **x*a = Kpub mod p** and now x represents private key in this.

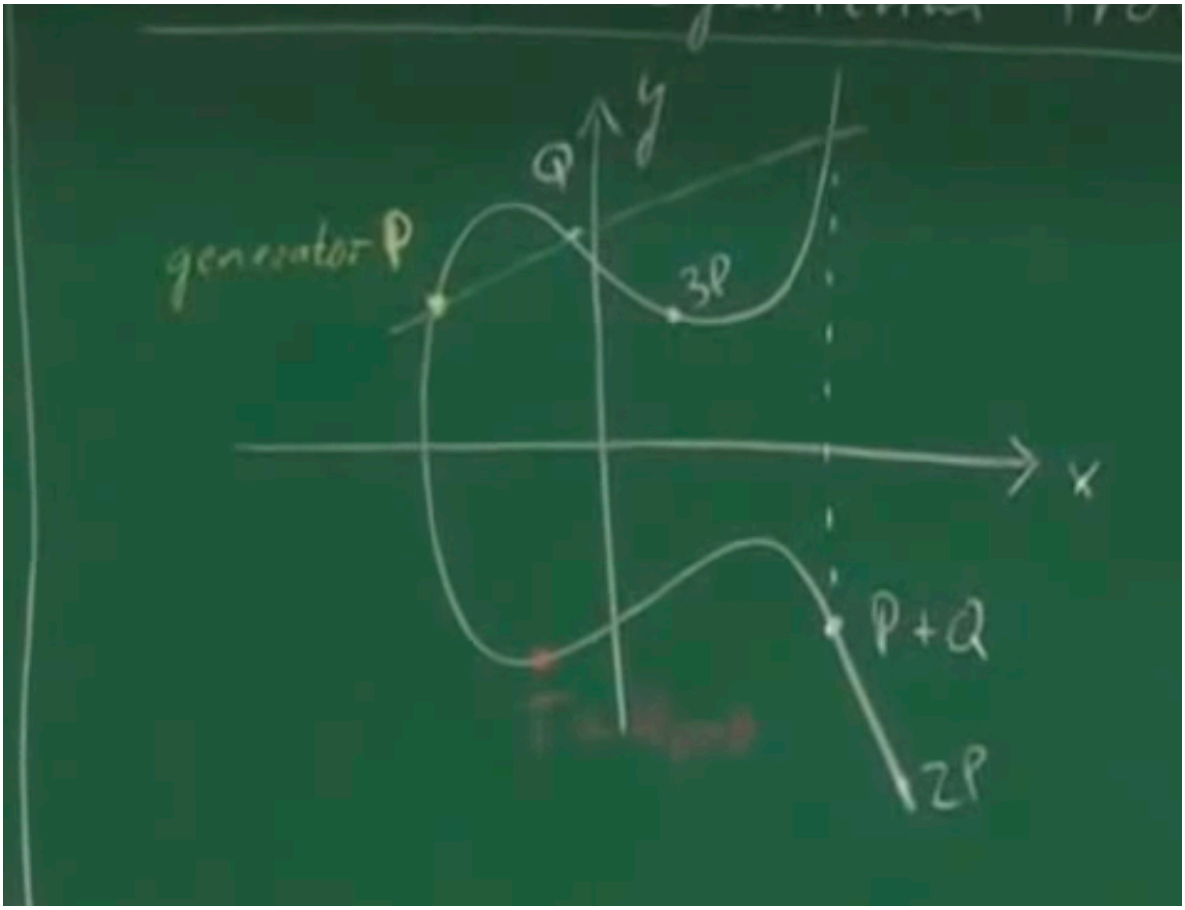
For now removing Mod p
In EC definition say x = d and generator is P then equation comes as d*P = Kpub and we got the DLP of EC.

Definition 9.2.1 Elliptic Curved Discrete Logarithm Problem (ECDLP)

Given is an elliptic curve E. We consider a primitive element P and another element T. The DL problem is finding the integer d, where $1 \leq d \leq \#E$*, such that:*

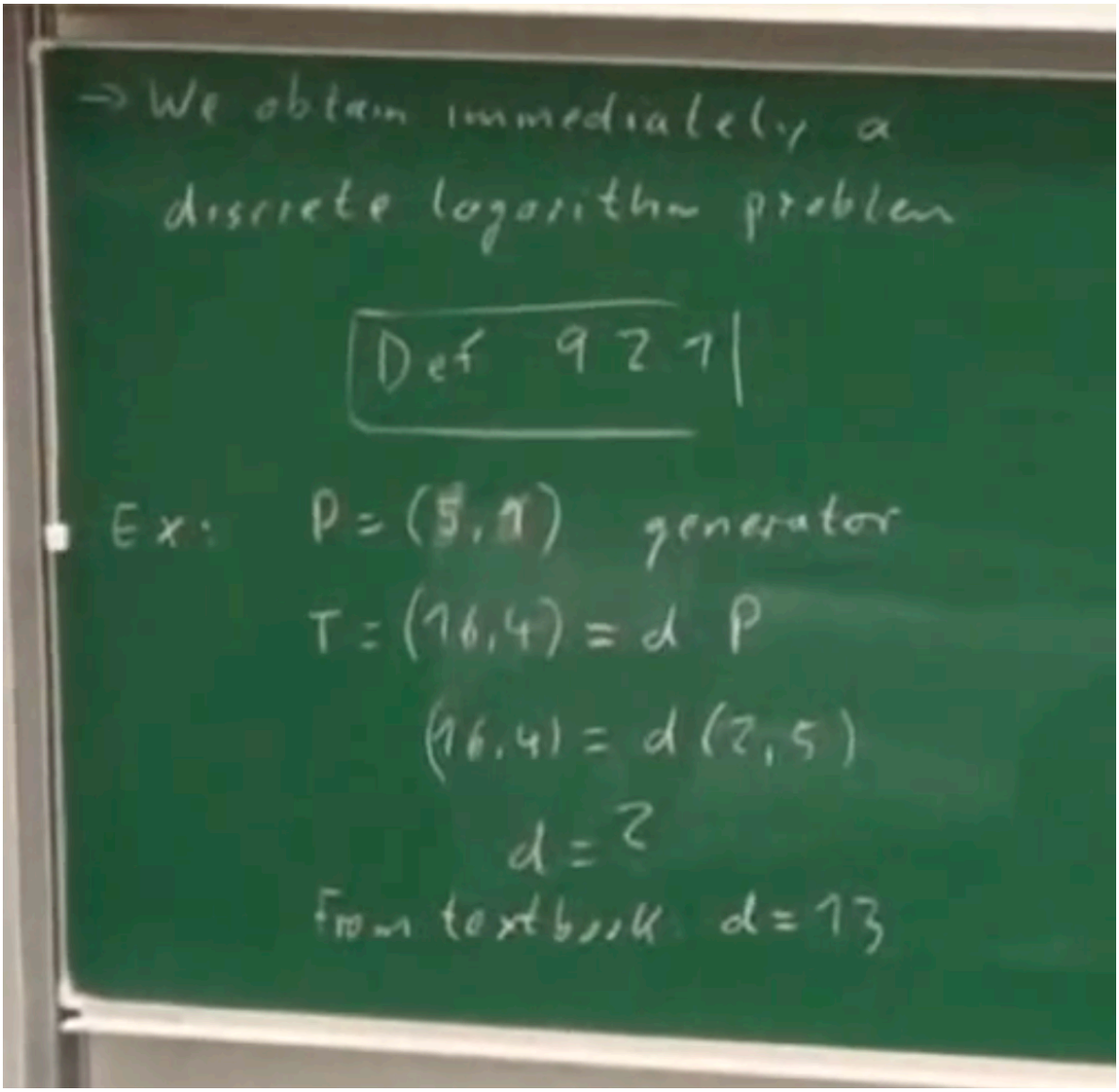
$$\underbrace{P+P+\cdots+P}_{d \text{ times}} = dP = T. \tag{9.2}$$

DLP problem is given P, given T find d ?
Graphical representation of EC DLP :



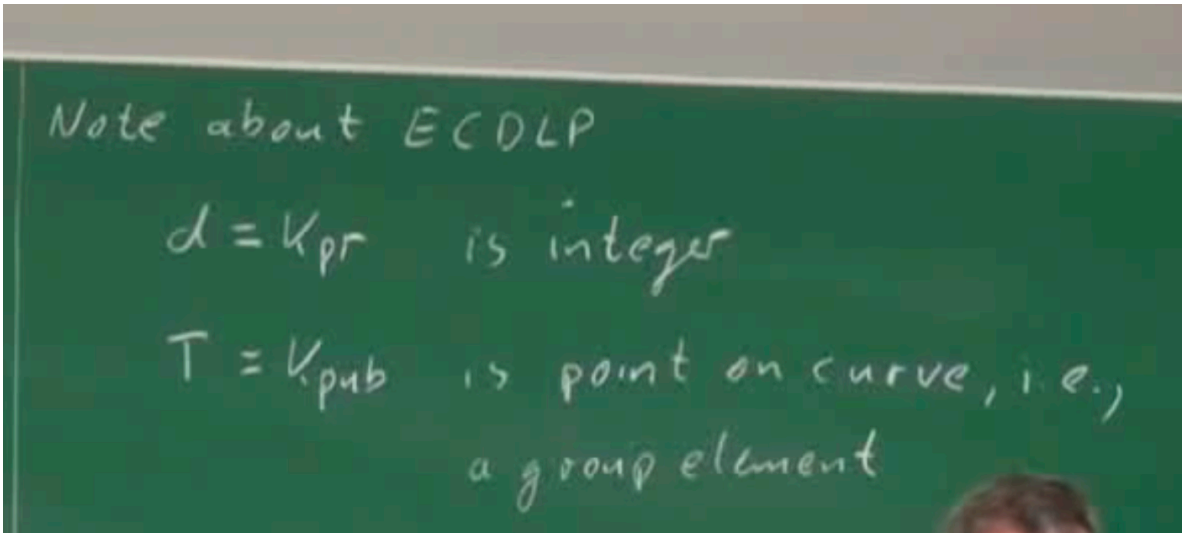
So we are given P and T and we need to find the number of hops to reach T and Number of hops in d which is Private Key.

Example of DLP:
say P = (5, 1) T = (16,4) P is the generator and we know T = d*P is possible.
Question is what is “d” ?
It is quite tough to find d, we need to compute every point.
From book: we found that d = 13.

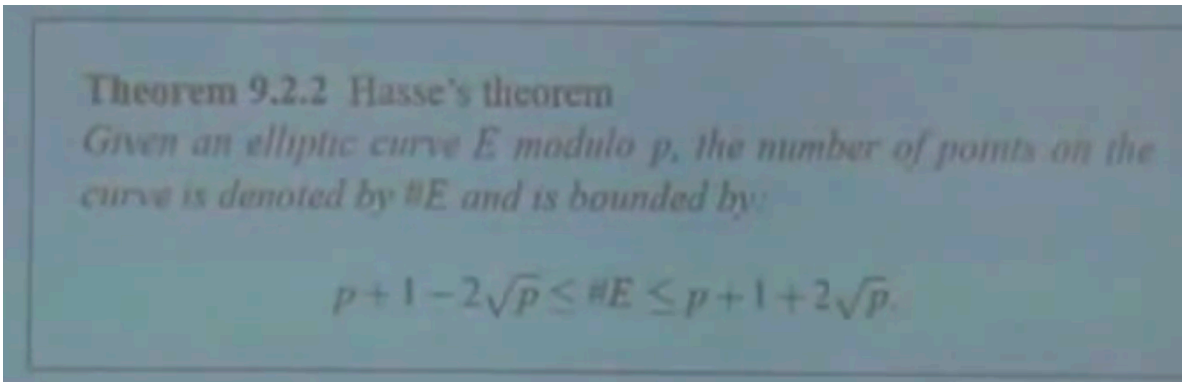


Note about EC DLP:
d is the Private key and what is the datatype of d ? it is integer, it is number of hops. it is not weird and this is true for all the DLP.

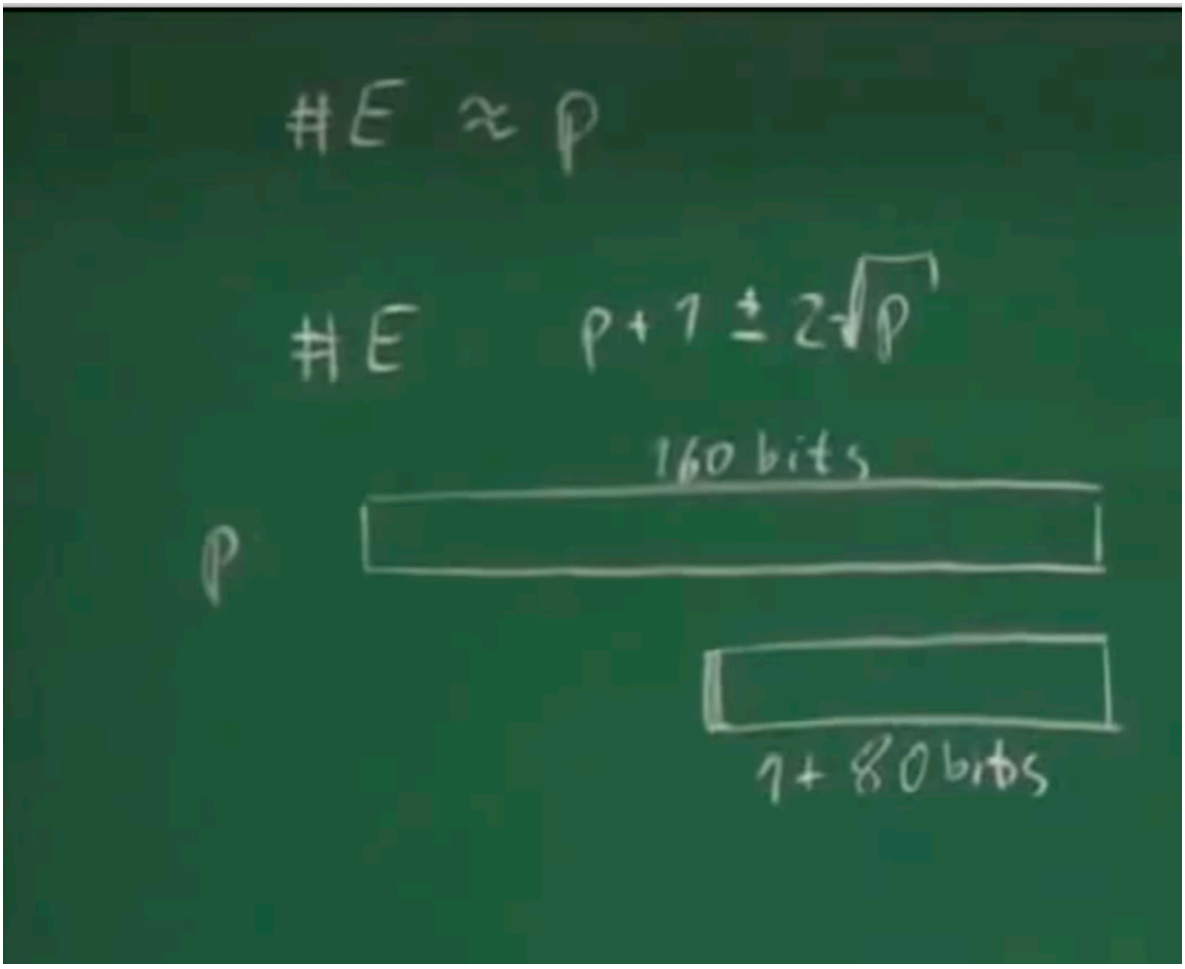
T is the Public Key and what kind of datatype is it ? it is a Point on the curve ie a group element.



Question: Group cardinality of EC or Size of group or Points in cyclic group or order of EC ?
Answer: In previous EC, cardinality of 19 where 19th point is Neutral element or Infinity point.
There is an algorithm by german mathematician



Hasse theorem gives us upper bound and lower bound.
it says that #E is in the range of **P + 1 +/- 2*sqrt(P)**
if we look into the formula we can say the upper bound and lower bound are gigantic in the sense if P is 2^160 but relative to P it is small.



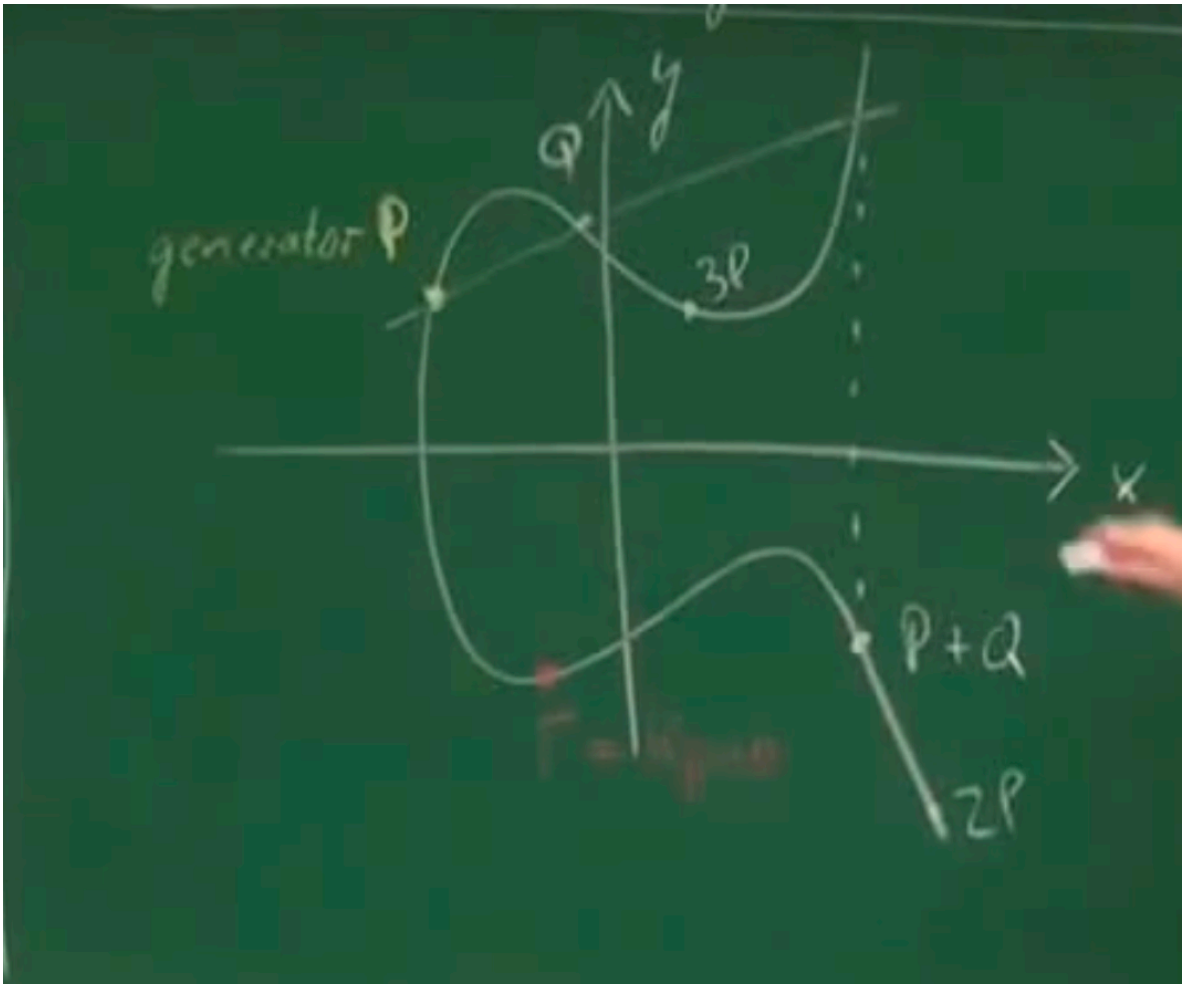
it is like 1 million you won and correction term is 1000.
if P ~ 2^160 Absolute error value is very big however relatively it is not big.
As per Paar, One needs the exact number of points in order to Thwart ie Prevent certain attacks.
Finding number of points exactly is computationally difficult so very often in practise people use standardised Elliptic curves, like NIST standard which we call NIST curve.

For those curves we know the exact #E.

All EC protocols rely on the hardness of the EC DLP.

Question: What does that mean ?
If we want to break Diffie Hellman/EC DH/ECDSA we need to solve this problem and we think that it is a hard problem.

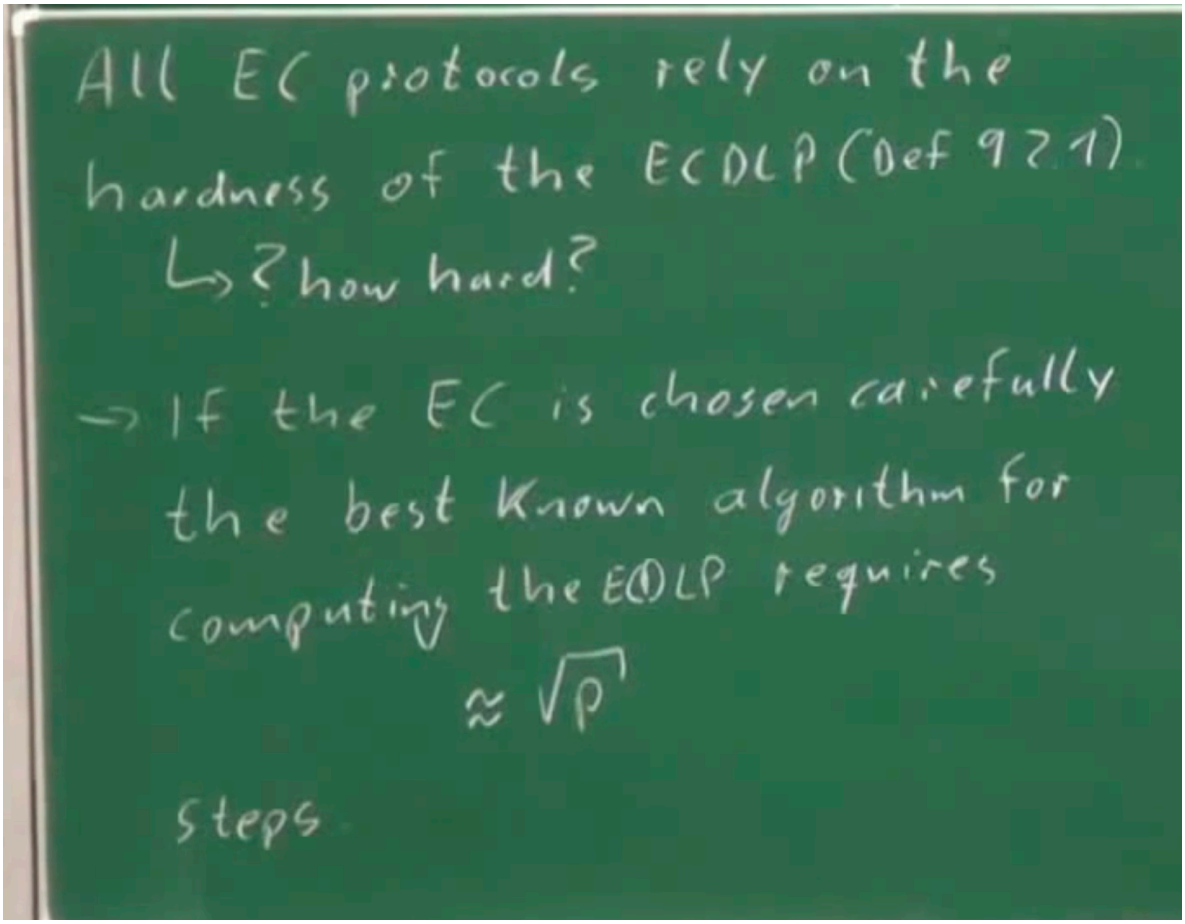
Question: As we know the EC ahead of time and also the primitive element that can make it vulnerable to bruteforce attack ahead of time ?
Answer: Yes of course, we are given everything and does that helps us? not it doesn't.



These curves are given but we need to choose the “d” ie private key which is taken randomly and then you can build the crypto system. you cannot precompute all the points on the curve as it is quite big numbers.

However not sure someone can get the key if that is the starting “d” -> This needs to be researched.

Question: How hard is it ?
-> if the EC is chosen carefully, the best known algorithm for computing EC DLP is SQRT(P) number of steps.



ie for P ~ 2^160 then the minimum steps required are 2^80.

160 bit key means 15-20 years of security but commercial tools already moved to 192 or 256 bit key size.

Part 2: ECDH -> Elliptic curve Diffie Hellman Key exchange.
One of the best thing about DLP protocol is that all the protocols looks very much the same ie once you give me a cyclic group like we have EC, we can do exactly same thing what we did in DH.
we can also do elgamal encryption also with this.
—> StraightForward adoption of DH in Zp*

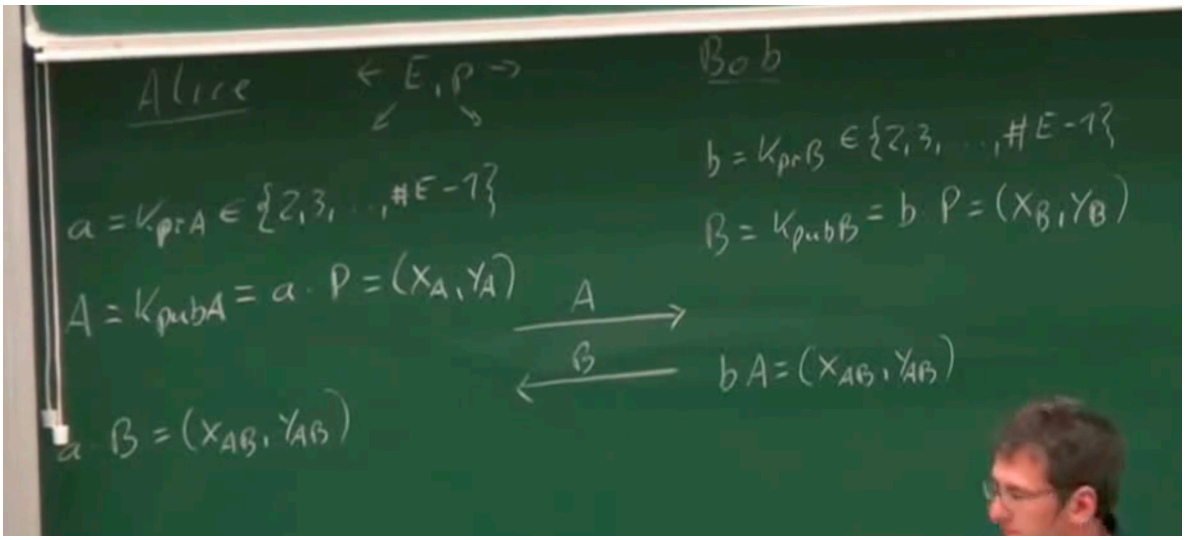
Phase 1: Setup, we need domain parameter, in Old DH we have a prime P and a generator but in EC we have domain parameters as:
E : Y^2 = X^3 + a*X + B mod P
primitive element P = (X1,Y1)

Phase 2: Protocol:
So we have the domain parameters now the same process as Generalized Diffie Hellman:

Alice computes private key, **Question is How ?** Answer is it is random.
Same Bob also computes private key.

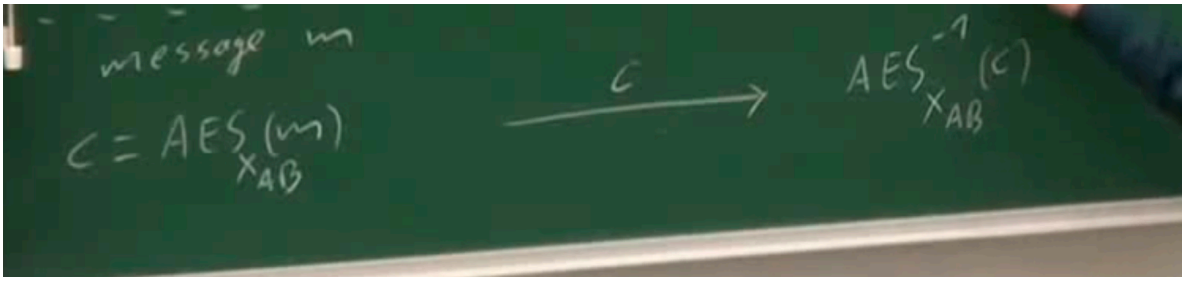
Question is what is the range of Kpr for both Alice and Bob?
Answer is {2,3,…., #E -1}
Now **Question is why not entire #E and 1 ?**
because with 1 P remains same and if we go to #E then you know that #Eth element is neutral element ie infinity and we don't want that to happen.

then alice and bob will compute public key using Kpub = a*P = (Xa, Ya) point on the curve.



Now question comes is how we can encrypt using "Point" :(
Actually in practise any of the coordinate can be chosen to encrypt ie either of the coordinates. Generally we take the X coordinate.

In case of AES, we need 128 bits only but X/Y coordinates are 160 bits, so we can take the leading 128 bits from the key but in practise we take the "HASH"



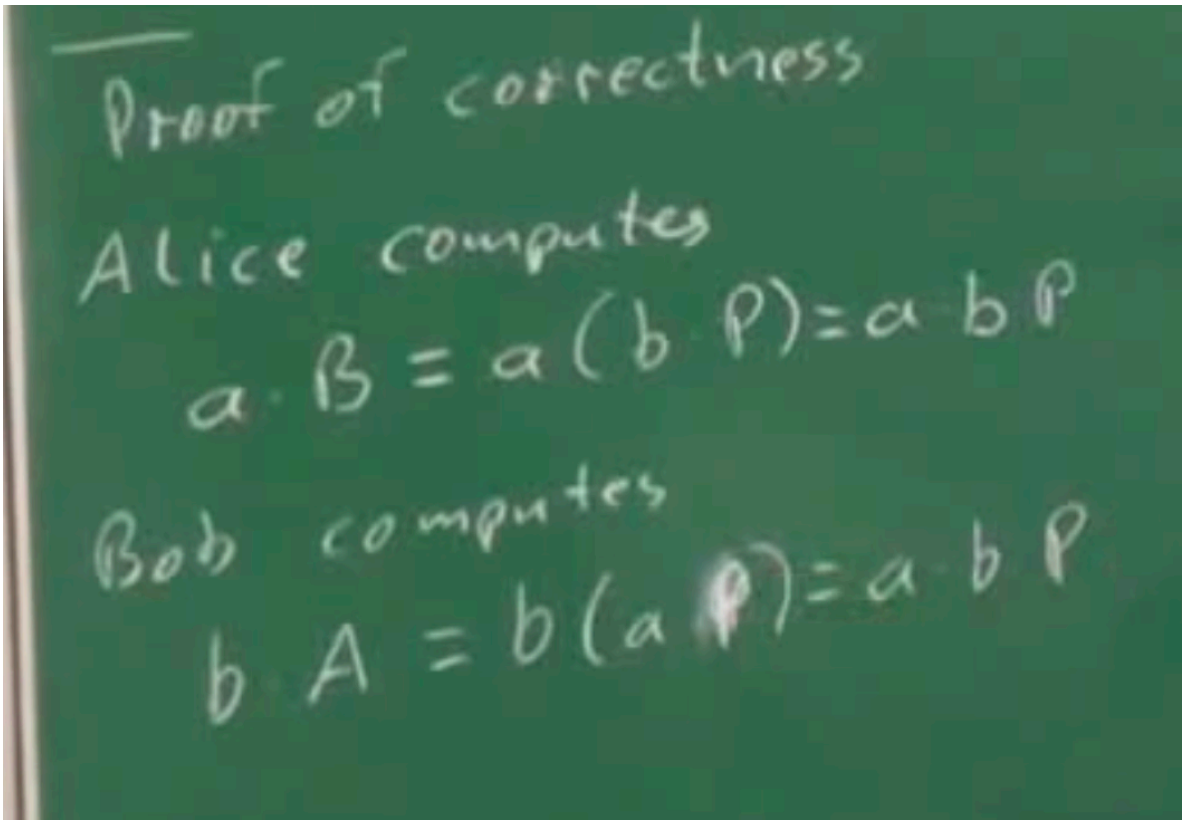
Lets look at an example:

Example 9.8. We consider the ECDH with the following domain parameters. The elliptic curve is $y^2 \equiv x^3 + 2x + 2 \bmod 17$, which forms a cyclic group of order $\#E = 19$. The base point is $P = (5,1)$. The protocol proceeds as follows:

Alice		Bob
choose $a = k_{pr,A} = 3$		choose $b = k_{pr,B} = 10$
$A = k_{pub,A} = 3P = (10,6)$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B = k_{pub,B} = 10P = (7,11)$
$T_{AB} = aB = 3(7,11) = (13,10)$		$T_{AB} = bA = 10(10,6) = (13,10)$

The two scalar multiplications that each Alice and Bob perform require the Double-and-Add algorithm.

Proof of correctness is simple so not writing that here.

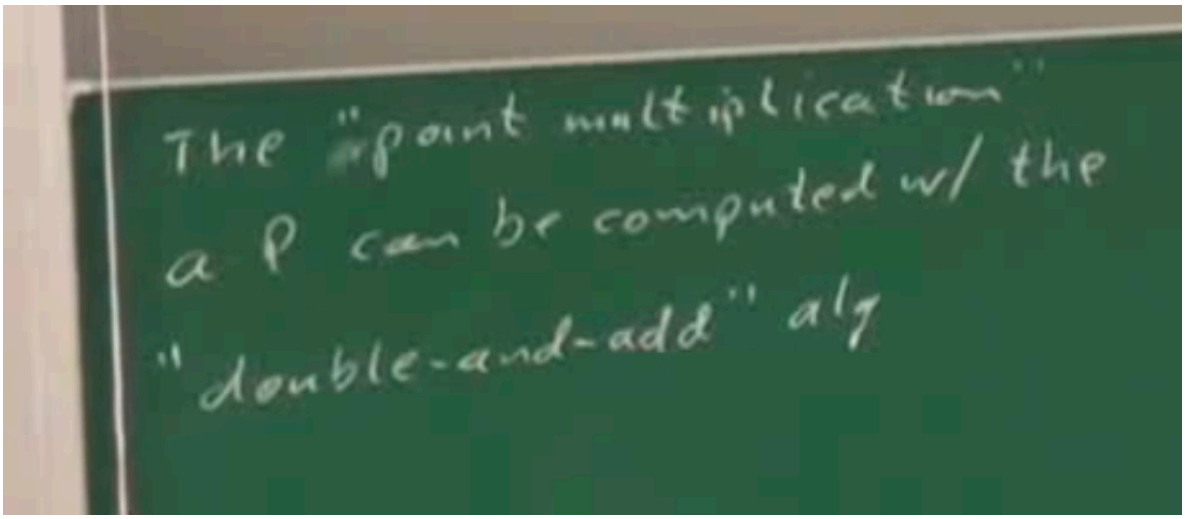


Computational Aspects:
What actual kind of operations do Alice and Bob have to do ?
Alice and Bob have to do group operations ie alice need to compute a*P and then a*B (Public key of bob)

Question is: How to compute the a*P ? Problem is that there is no direct way to compute a*B ie we have to do a+a+a+a... p times.

but if we have really large numbers how to do this ?
In case of number^number we are using Square and Multiply algorithm however in this case also we are going to do the same.

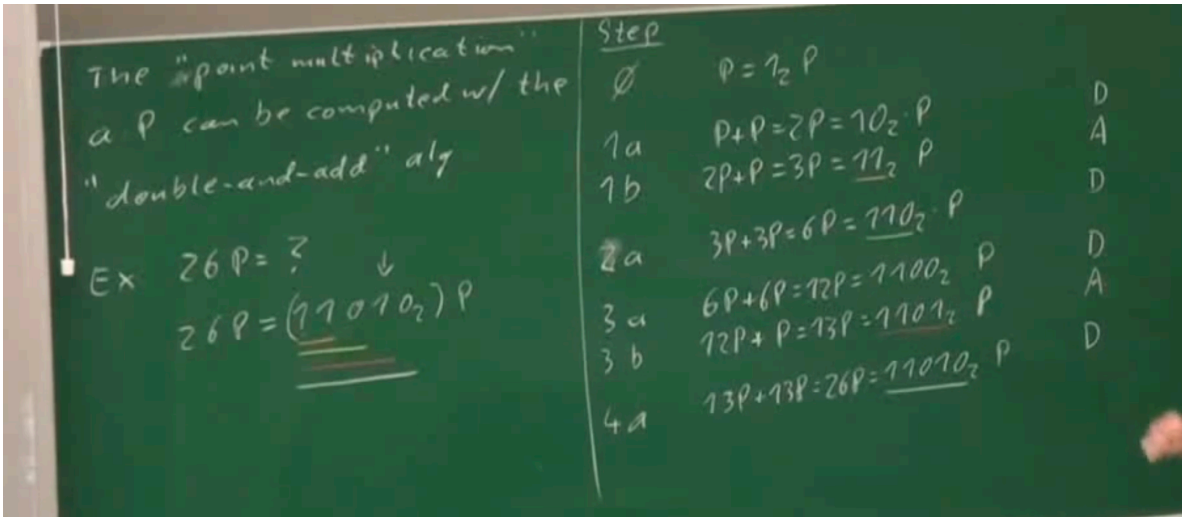
Squaring in EC becomes P + P and Multiply becomes add
ie it becomes Double and Add algorithm.



eg: 26 P ?
we can do P + P + ... 26 times
instead we do here is double and add algorithm.
so we need to find the binary representation of 26 = (11010)P
we will start scanning from left to right.
1 -> Double and Add except for 1st bit.
0 -> Double

1 -> P
1 -> P + P = 2P , and now we need to add 2P + P
0 -> 3P + 3P = 6P
1 -> 6P + 6P = 12P , and now add = 12P + P = 13P
0 -> 13P + 13P = 26P and we are done.

or we can do one more way:
1 = P
10 => P + P = 2P
11 => 2P + P = 3P
110 => 3P + 3P
1100 => 6P + 6P = 12P
1101 => 12P + P = 13P
11010 => 13P + 13P = 26P


The "point multiplication"
a P can be computed w/ the
"double-and-add" alg
Ex 26P = ?
26P = (11010₂)P
Step
∅ P = 1₂P
1a P+P = 2P = 10₂P D
1b 2P+P = 3P = 11₂P A
2a 3P+3P = 6P = 110₂P D
3a 6P+6P = 12P = 1100₂P D
3b 12P+P = 13P = 1101₂P A
4a 13P+13P = 26P = 11010₂P D

Done.