

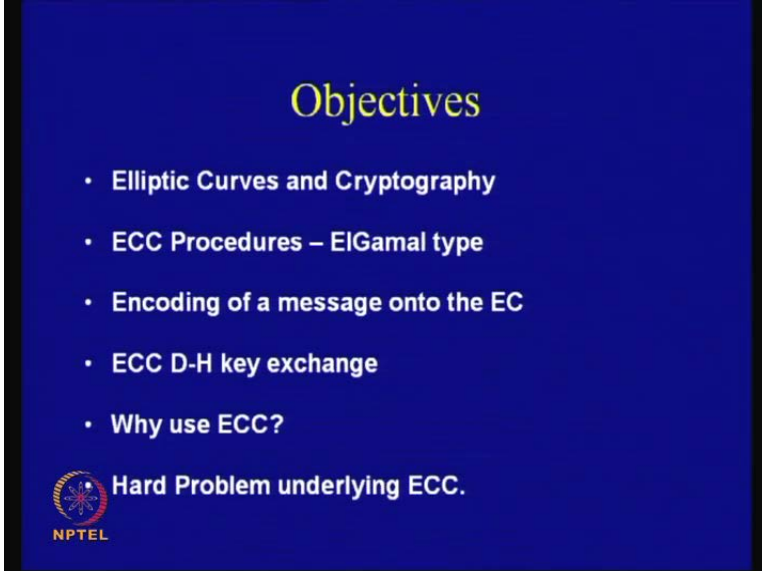
Cryptography and Network Security
Prof. D. Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No. # 01

Lecture No. # 35


Applications of Elliptic Curves to Cryptography

(Refer Slide Time: 00:33)



Objectives

- Elliptic Curves and Cryptography
- ECC Procedures – ElGamal type
- Encoding of a message onto the EC
- ECC D-H key exchange
- Why use ECC?

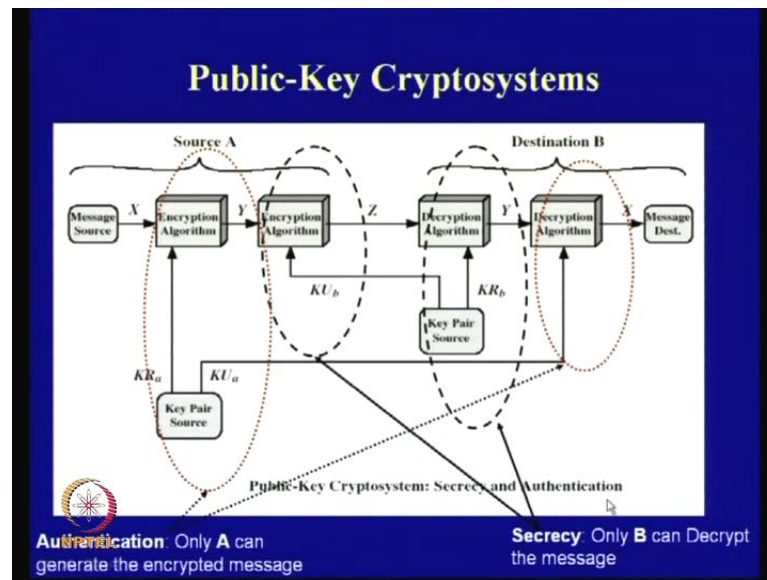
 **Hard Problem underlying ECC.**

In the last class, we have studied about elliptic curves and their definitions. So, in today's class, we shall concentrate on how to apply elliptic curves to cryptographic operations. In today's class, we shall essentially cover these areas like, what is the relation between elliptic curve and cryptography, discuss about an El Gamal type of encryption algorithm in context to elliptic curves. The other important thing that is needed is how essentially do we take a message and encode that into the elliptic curve. So, we will see a very simple algorithm to do so, and also the corresponding decoding, that means, given the corresponding point how we get back the message.

Then we shall discuss about the Diffie-Hellman key exchange in context to elliptic curves, and then, we shall address that why do we use elliptic curves, because we have

studied RSA and public key ciphers which are based on discrete logarithm problems. So what is the purpose of elliptic curves and what is the underlying heart problem behind elliptic curves, we shall just discuss certain issues in this context.

(Refer Slide Time: 01:26)



So, first of all to start with public-key ciphers we have seen there are two important requirements, one is secrecy, and other one is authentication of information. As we know that in any public-key cryptosystem these are provided by a pair of keys, so if you know that if you want to apply your public-key cipher for giving you secrecy, then also you use a pair of public key and private keys. Similarly, if you want to use it for authentication you use another pair of public key and private keys here.

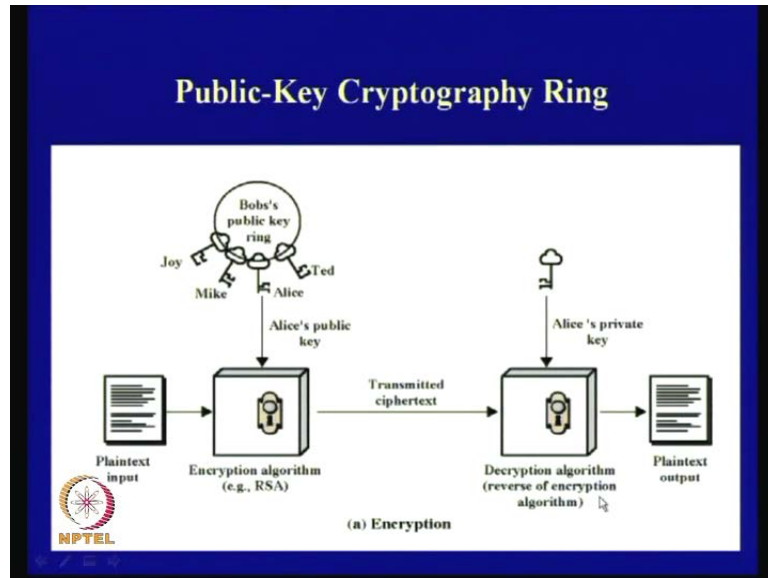
The only difference when you are using it in context to security, and when you are using it in context to authentication is that, how do you use I mean, which key do you use for what reasons. For example, when you are interested in or motivated in the security of or the secrecy of the information then you use the public key. For encrypting, the sender uses the public key, but the receiver decrypts it using the private key. Similarly, when you are using it for authentication you can use the same public key, but you have to use it in the other way that is, if you want authentication then you have to use the private key for sending the data and use the public key for verifying. So similar to that we can also use, as shown in (refer slide time: 01: 26) that there is a source A, a destination B, and the message source wants to send X which is a message, and it wants to send it to the

destination ensuring that both secrecy of the information and the authentication of the information is maintained. So, for source A chooses a pair of messages I mean pair of keys, and for encryption algorithm there are two stages of the algorithm, one is the public key encryption used for secrecy, and other one is used for authentication.

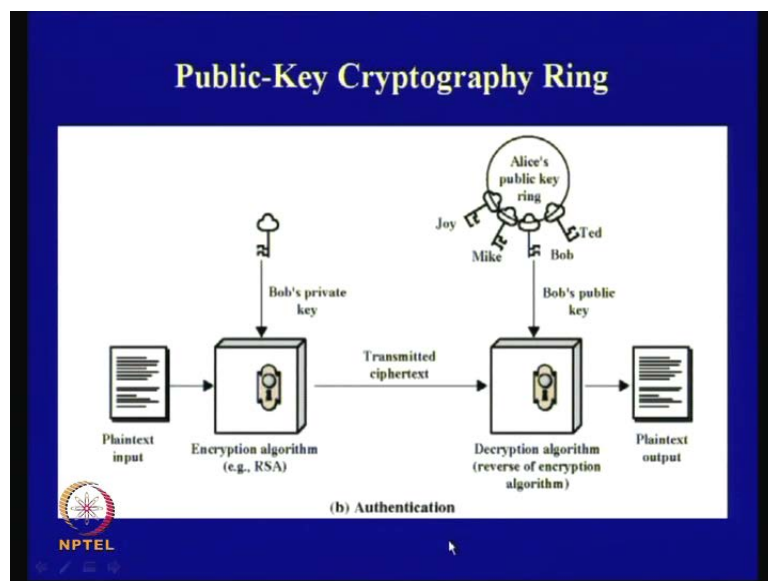
When we are using for encryption **then you are using the public key** because you have to encrypt the data. So, you essentially choose the public key and you encrypt the data and send it to the receiver, but for authentication you do just the opposite that is you essentially choose the private key for sending, and similarly if you want to verify you use the corresponding private key. Therefore, if you want to send it for authentication then you see that you have essentially chosen the public key, for performing your authentication **so that the private key for performing or signing the data**, and for verifying you use the corresponding public key. But if you want to use it for secrecy of the information then you use just the keys **but in the same keys or similar pair of keys**. But in the other way that is what you do is that, you choose the public key for performing your encryption operation, and you decrypt the corresponding cipher-text by using the corresponding private key right, because anybody should not be able to decrypt.

So therefore, the secrecy of the data is ensured because you know that the secret key is **only that is the assumption there is the secret key only B has the corresponding secret key therefore or the private key** and therefore it can only decrypt. And similarly, the authentication is ensured, because if the verification is successful it shows that based on the assumption that only A has the corresponding private key, so he knows that this particular message digest can be only generated by the source A. **that is since the I mean** apart from source A nobody has the corresponding private key, hence only A can sign this message. Therefore the authentication is also ensured in this fashion.

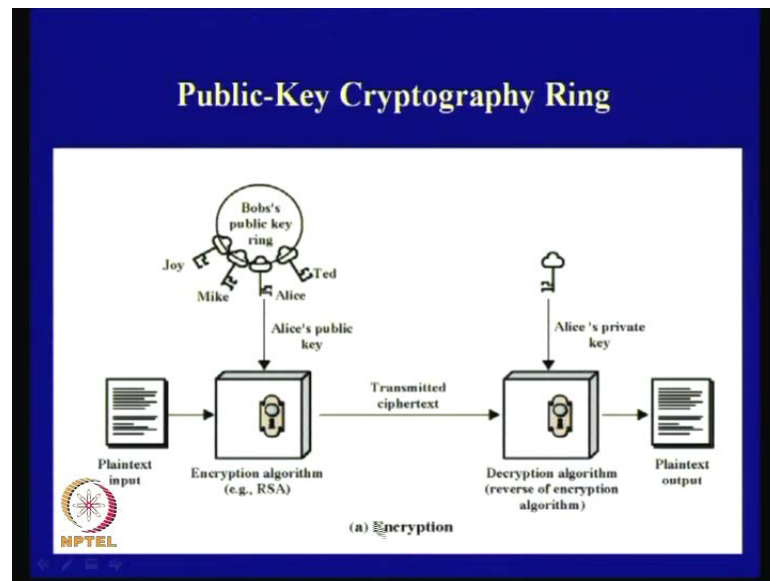
(Refer Slide Time: 05:24)



(Refer Slide Time: 05:30)

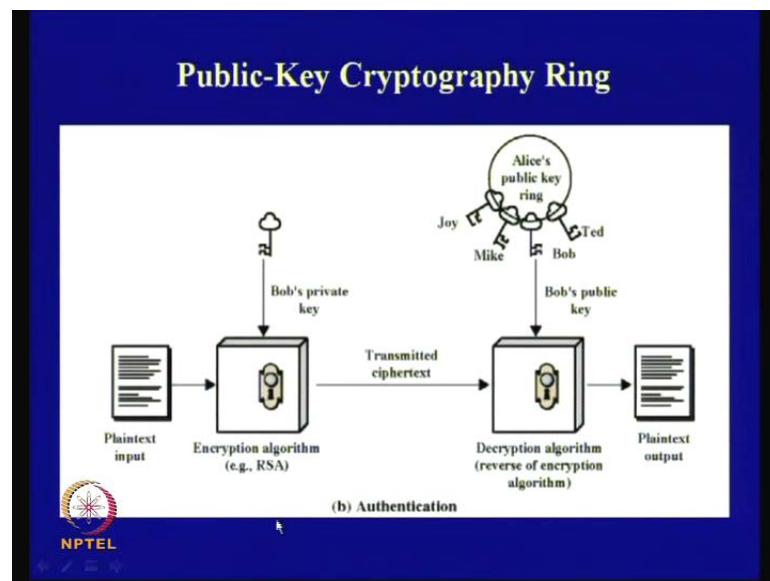


(Refer Slide Time: 05:34)



So this is the very customary operation of how you basically there are two important parts; one is the encryption, and the other one is the authentication. So, what we understand from here is that, what we have studied previously is that you can use any encryption or any public key encryption algorithm, and we can actually achieve this goal of encryption as well as authentication.

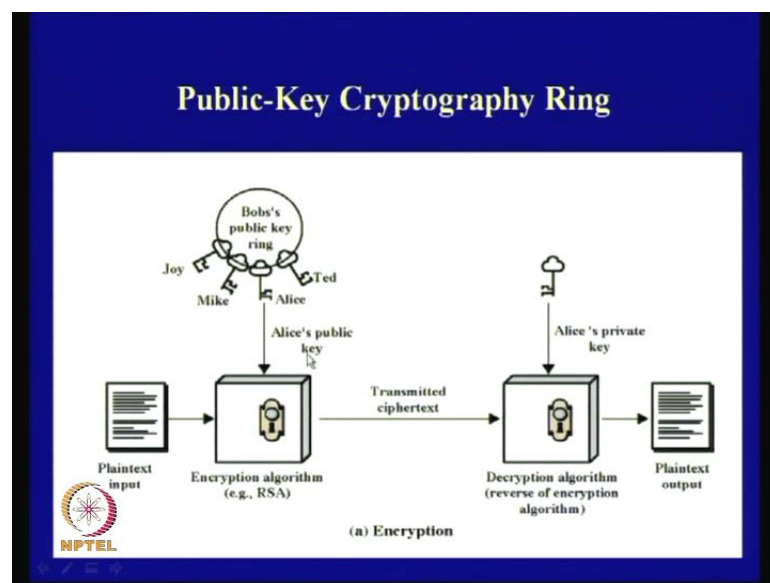
(Refer Slide Time: 06:13)



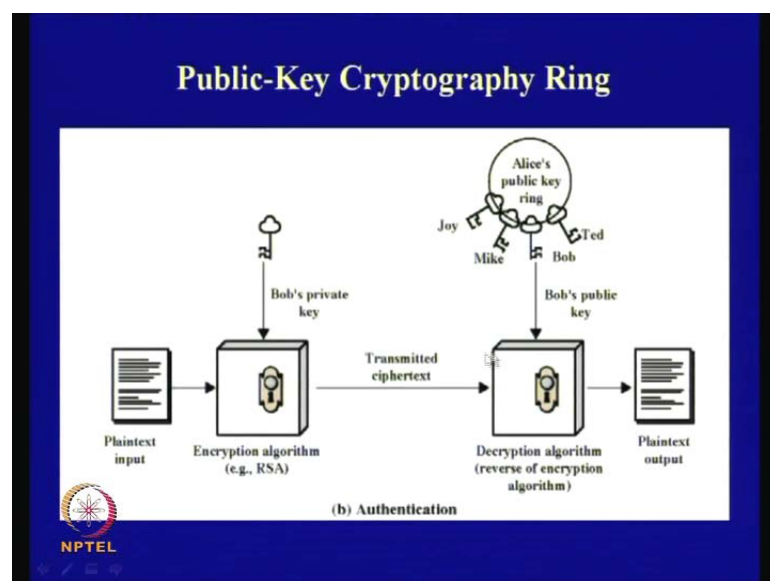
So, for encryption there is a public key that is a public key ring, so if you want to communicate with anybody for example, if you want to communicate with Alice then

from the public key ring you choose the corresponding public key for Alice and you generate the ciphertext, and the receiver since it has the corresponding private key decrypts the message and obtains the corresponding plain text. When you are doing the authentication, then basically you are generating the signature, so you take an input and you sign it by your own private key and you send it, and if the receiver is expecting the message from Bob then it chooses the corresponding public key from the public key ring and decrypts it and obtains a plain text message.

(Refer Slide Time: 06:41)

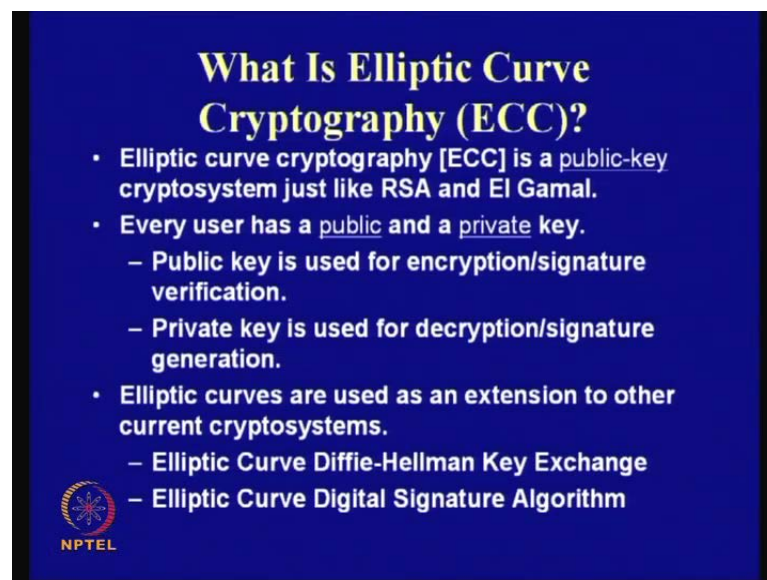


(Refer Slide Time: 06:51)




So you see that both these essentially uses the same public key cryptography ring, but the point to be noted here is that when you are using it for encryption, and when you are using it for authentication the public key ring is, in the first case it is present in the sender's part and in the second case it is present in the receiver's part. So that is the small thing which is to be kept in mind. Now there are various arguments about whether you should do the encryption first or whether you should do the authentication first, so that essentially is a matter of we can argue on that. But this is the basic way how we can actually choose the public key, and you can actually perform either if you want to use it for the secrecy or you want to use it for the authentication of the information.

(Refer Slide Time: 07:21)



What Is Elliptic Curve Cryptography (ECC)?

- Elliptic curve cryptography [ECC] is a public-key cryptosystem just like RSA and El Gamal.
- Every user has a public and a private key.
 - Public key is used for encryption/signature verification.
 - Private key is used for decryption/signature generation.
- Elliptic curves are used as an extension to other current cryptosystems.
 - Elliptic Curve Diffie-Hellman Key Exchange
 - Elliptic Curve Digital Signature Algorithm

 NPTEL

Therefore what is elliptic curves, we have studied that elliptic curves is essentially I mean, elliptic curve is basically a certain set of curves that is essentially what we have studied in the last class. Now the question is what elliptic curve cryptography is, so elliptic curve cryptography is nothing but a public key cryptosystem just like we have studied RSA and El Gamal. So it also has a public and a private key, and the public key is used either for encryption or the signature verification, and the private key is used for decryption or for signature generation. So elliptic curves are used as an extension to other current cryptosystems therefore, you can also have elliptic curve Diffie-Hellman key exchange, you can have elliptic curve digital signature algorithms. Wherever you have seen the previous applications of public key ciphers, you can have similar

applications here also, but these are actually based on certain curves which are known as the elliptic curves which are having cubic curves, which we have studied in the last class.

(Refer Slide Time: 08:19)



Using Elliptic Curves In Cryptography

- The central part of any cryptosystem involving elliptic curves is the elliptic group.
- All public-key cryptosystems have some underlying mathematical operation.
 - RSA has exponentiation (raising the message or ciphertext to the public or private values)
 - ECC has point multiplication (repeated addition of two points).

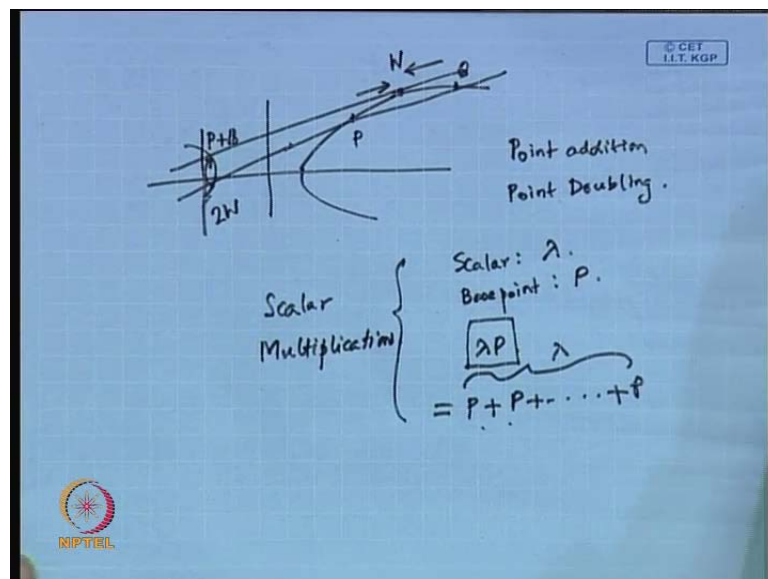
 NPTEL

Now, we shall study that how do we apply this elliptic curves to generate a public-key cipher. So the central part of any cryptosystem which involves elliptic curves is the elliptic group. We have studied that, if you want to generate elliptic groups then **for that in order** to do that we actually define certain operations, the operations means we take two points on the elliptic curve and we define what is mean by addition on these two points. So we define what is meant by point addition and what is meant by point doubling.

So, all public key cryptosystem have some underlying mathematical operations like, if you want to for example, chose RSA then you have exponentiation, you are basically doing a modular exponentiation, and all public-key ciphers essentially relies upon certain things which are assumed to be one way function that is, some multiplication easily compute what which are difficult to invert. So, in context to RSA we have seen that the heart problem or the one way problem was essentially the factorization problem that is, if **I can take two if** there is a large product or if there is a large composite number which can be factored into two large prime numbers, then it was believed that it is difficult to factorize these module. That was the assumption on which the RSA security was derived. So when we studied about the El Gamal cryptosystems which are based upon the finite

field for example, F_p or any other finite field for that matter, then it is based on something which is called as a discrete log problem. So it was known we leave it is easy to compute given a public module g , and given a secret module x , it is easy to compute g power of x modulo P . But from g power of x modulo p , it is actually believed to be difficult to compute the value of the exponent x , so that was the assumption on which the El Gamal cryptosystem was based on. So similarly, here in elliptic curve cryptography we also have something which is then similar or analogous to the discrete log problem, and which is called as the elliptic curve discrete log problem. So this essentially relies on how do we do the point addition, and how do we do the point doubling. So the underlying mathematical operation on which this public cryptosystems based on elliptic curves are actually something which is called as point multiplication, so we take an elliptic curve for example, consider any elliptic curve for that matter.

(Refer Slide Time: 10:52)




Suppose, this is one elliptic curve, we take a point p on this curve, and we take another point Q on this point and we do a point addition, so that is what we have studied in the last class. Therefore, these essentially intersect the curve at the third point and the corresponding reflection of that point is the sum O plus Q , so this is known as the sum P plus Q . Similarly, if this P and Q points converge and there is only one single point, we draw a tangent at that point and we reflect that point, so that was called the doubling operation. Therefore if this P and Q converge at a point call it say W , then this is nothing but twice of w because that is W plus W , so this is basically the operation of point

addition and point doubling. Now, in elliptic curve cryptography what we essentially do is that we choose a scalar quantity say call it lambda, and there is a base point which we call as a point P. So the whole point is how do we compute lambda into P, so that is the basic operation on which elliptic curve cryptography relies that is how do we compute these value of lambda multiplied by P. So, one obvious way of computing lambda into P is by taking P and by adding it lambda times, so we take P and we keep on adding them lambda number of times. So we can take a point P and P plus P Means, it is a doubling operation on P then whatever output we get again we add that with P and we keep on doing that lambda number of times. So that is essentially something which is called as the scalar multiplication, which is central to what is known as elliptic curve cryptography. So whether we want to use it for encryption, whether we want to use it for key exchange, this is the most important and center operation on which elliptic curve cryptography is based on.

(Refer Slide Time: 13:08)

Generic Procedures of ECC

- Both parties agree to some publicly-known data items
 - The elliptic curve equation
 - values of a and b
 - prime, p
 - The elliptic group computed from the elliptic curve equation
 - A base point, B , taken from the elliptic group
 - Similar to the generator used in current cryptosystems
- Each user generates their public/private key pair
 - Private Key = an integer, x , selected from the interval $[1, p-1]$
 - Public Key = product, Q , of private key and base point
 - $(Q = x \cdot B)$

 NPTEL

So, the generic procedure of doing elliptic curve cryptography is that both parties agree to some public known items. For example, the elliptic curve equation as we have studied, are a form of $y^2 = x^3 + ax + b$. Similarly, we have got generalized Weierstrass equation, so if I give you the value that is the values of the constants, then you essentially know the corresponding curve equation. So it is believed that everybody is a sender and the receiver, and even the adversary knows the curve equation. Now, you have to basically define elliptic group, so elliptic group is computed from the elliptic

curve equation and a base point B which is from the elliptic group, and similar to something which is called as what we have seen as generator used in the context of the previous public key ciphers. So similar to that, we have got a base point B which is again a public domain information. These are the public domain information, the values of a and b . So what is a and what is b , this you will remember from the Weierstrass equation $y^2 = x^3 + ax + b$, therefore those two a and b will be important in understanding what is the corresponding curve. So based upon that we have got various varieties of curves, we have got random curves, and there are other forms of curves.

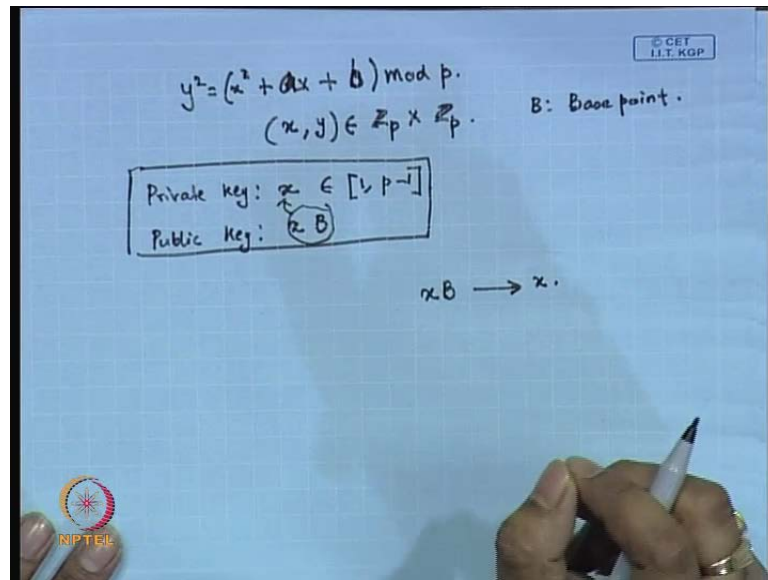
Similarly the prime p , the corresponding elements of the elliptic curve is based on some field therefore, that field could be a prime field. So, in this case we are assuming that it is a prime field, it could be a characteristic to field also like g^f 2 power of n , and it could be essentially some complex sets also. But if I assume that for all purpose, let us assume that the underlying elements on that elliptic curve are chosen from F_p that is it is chosen from the field which is generated out of numbers from 0 to P Minus 1, where p is a prime number. So the elements on the curves are actually chosen from 0 to P Minus 1, and already I did not say explicitly all the number of points which are actually there on the elliptic curves, they are actually finite so that is basically a finite set of points which are there on the elliptic curve. Therefore, you have got your F_p which is your set of numbers from 0 to P minus 1, where p is a prime number, and then you have got some points which were actually choosing on the elliptic curve. So those are actually ordered pairs, like addition points x comma y , where X Also belongs to F_p , Y Also belongs to F_p .

So, basically it is a subset of the numbers from Z_p cross Z_p , it is a finite set, so it is a finite set of numbers and finite set of points which actually are there on the actual elliptic curve. And by our previous construction of the addition, and the definition of the addition operation on the points, those elements x comma Y actually form a mathematical group along with the point on infinity. Therefore, if I take all these points x comma y which actually satisfies this Weierstrass equation, those points along with the point on infinity form what we call the mathematical poof under the definition of the addition and the doubling operation.

So therefore, in this case the public known data items are the values of a and b , and the corresponding prime p , these are public numbers. And similarly, there is a base point which is also a public domain value. Now, each user generates their public or private key

pair, so what is a private key pair here, it could be an integer x which is selected from the interval from 1 to P minus 1. So from 1 to P minus 1 the private key is any integer x which is selected at random, and the public key is actually the product of x and $P B$.

(Refer Slide Time: 17:14)

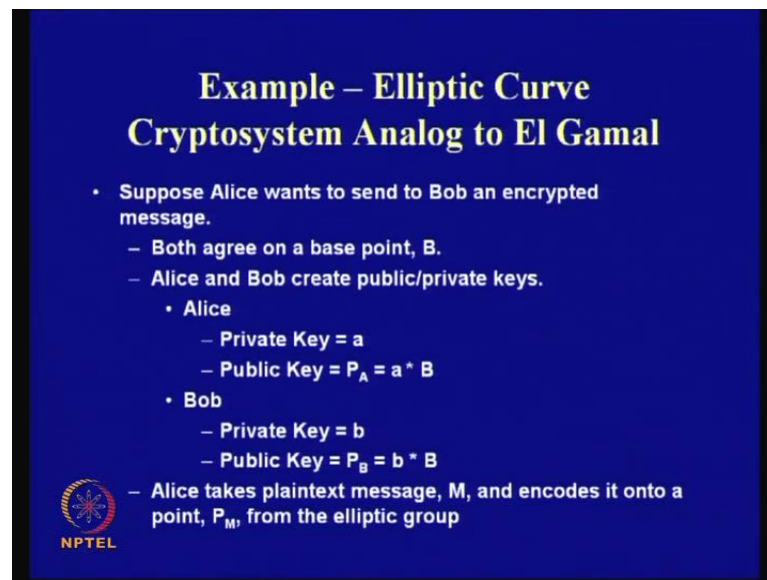


Therefore, if you want to use it you first of all choose a corresponding curve equation. So y equal to x cube plus $a x$ plus b , and whenever you are doing it there is a modulo P , because this elements x and y are essentially chosen from \mathbb{Z}_p cross \mathbb{Z}_p , and the corresponding private key which we are choosing is another number. So, although it is x it is a different x , actually it is the secret x which is chosen from the set closed interval from 1 to P minus 1. And your public key is actually x multiply by the base, so B again another public parameter which is the base point of the curve which is generally provided by a central body. So for example, we have got mixed curves, therefore they essentially gives you this values of a and b , I think it is better to make this a and b as small a and small b is generally provided by a central body, it is a chosen curves. So this is the way how you choose or calculate this private key and the public key.

So, immediately you know that from our previous discussions if this x number is chosen from 1 into P minus 1, and this public key value is public therefore, everybody knows this x into B . So, whenever you are using it for cryptography, it is obvious that one should not be able to know this private key. Therefore, it becomes immediate that from this $x B$ it should be difficult to get the knowledge of x , which is the basic assumption.


So, how do you get x into B that is by adding B x times, repeated additions will give you x into B , and this problem of getting x from $x B$ is the basic assumption. **again the central believe to be mathematically difficult problem** on which elliptic curve cryptography relies upon.

(Refer Slide Time: 19:33)



Example – Elliptic Curve Cryptosystem Analog to El Gamal

- Suppose Alice wants to send to Bob an encrypted message.
 - Both agree on a base point, B .
 - Alice and Bob create public/private keys.
 - Alice
 - Private Key = a
 - Public Key = $P_A = a * B$
 - Bob
 - Private Key = b
 - Public Key = $P_B = b * B$
 - Alice takes plaintext message, M , and encodes it onto a point, P_M , from the elliptic group


 NPTEL

So, anyway I will come to that in more details, so at least first of all see that how do we do the basic encryption operation. Suppose Alice wants to send to Bob an encrypted message, so both agree on a base point B , and creates a public private key. So what Alice does is that, it chooses a private key a and the corresponding public key is P_A , which is $a B$ that is being added a number of times. Similarly, Bob's private key is again small b and the corresponding public key is b multiplied by B . Now, this public keys are actually available to everybody, it is shared in the public key ring, so if Alice wants to essentially encrypt a message M , then it has to somehow convert this message or encode this message into a point onto the curve so you call it P_M .

(Refer Slide Time: 20:45)

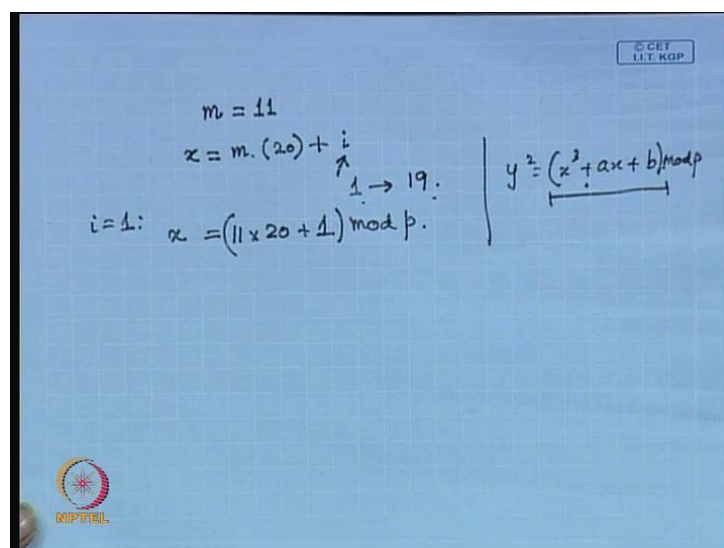
Encoding of a message onto the Elliptic Curve

- Consider a curve: $y^2 = x^3 + ax + b$
- The plaintexts are say numbers and English Characters (0-9 and 10-35).
 - 'B' is encoded as $m=11$.
- Choose a public variable, $k=20$.
 - compute, $x = mk + i$. Vary i from 1 to $k-1$ and try to get an integral value of y .
- Thus, m is encoded as (x, y) .
- The decoding is simple: $m = \text{floor}((x-1)/k)$.



Therefore, the first thing is that let us say, we can actually assume that it is taking this M , somebody can encode that into the point on the curve. Therefore, obviously one may ask that how do we essentially take this message M and encode them into the point P , so there are various ways but one very simple technique is shown here. You consider this curve y square equal to x cube plus a x plus b , so one way could be like this that is the plain text say numbers and English or roman characters, whatever you say from zero to nine, and from ten to thirty five. **suppose therefore** if I want to send this character B , then this B will be encoded as the number n equal to eleven.


(Refer Slide Time: 21:43)



© CET
I.I.T. KGP

$$m = 11$$
$$x = m \cdot (20) + i$$

$i \rightarrow 1 \rightarrow 19$

$$i = 1: x = (11 \times 20 + 1) \bmod p.$$
$$y^2 = (x^3 + ax + b) \bmod p$$



So, these are very simple way of doing, you see that you choose a public key variable say k equal to twenty, and it start computing x which is equal to n into k plus i and it vary i from one to k minus one. Therefore, what I am doing is that, you know that this value that is k is a public known value, and you want to send the value m equal to eleven, that is what my **related** the character b indicates or is encoded. So I will start computing X As m multiplied by a publicly known value say call it twenty, and I start adding some value i to it, and I will vary this i from one to k minus one, k being equal to **twenty is** nineteen and I keep on varying this i at one step. So for example, in this case it is eleven into twenty plus one. To begin with when i is equal to one, so first of all in the first iteration i is one you get a corresponding value of x , you remember that you are always doing a modulo p . So there is some p on which you are doing a modulo p operation, let it be some value essentially.

So now, if you get this value of x there is a corresponding curve equation, so y square equal to x cube plus a x plus b . You take this x and you substitute that into y , and this is again a modulo p . Therefore, this number which we get should be a quadratic residue, because if it is so then there is an integral solution for y . So that you can check by **there are** techniques that we have seen previously, and you know how to check whether a given value is a quadratic residue modulo p or not. Therefore, you can check whether it is a quadrate residue modulo p , if it is so then x comma y is corresponding encoding, otherwise you again start incrementing this i to the next value two. Now it is quite easy, you can check that if I start varying from one to nineteen at least one of them we ensure that y is a quadratic residue. **so if I keep on increasing this like this there at least one of them will be ensure that it is a quadratic residue so that we can just think upon why it is**

(Refer Slide Time: 23:55)

Encoding of a message onto the Elliptic Curve


- Consider a curve: $y^2=x^3+ax+b$
- The plaintexts are say numbers and English Characters (0-9 and 10-35).
 - 'B' is encoded as $m=11$.
- Choose a public variable, $k=20$.
 - compute, $x=mk+i$. Vary i from 1 to $k-1$ and try to get an integral value of y .
- Thus, m is encoded as (x,y) .
- The decoding is simple: $m=\text{floor}((x-1)/k)$.



(Refer Slide Time: 24:16)

Example

- $p=751$, $a=-1$, $b=188$ and $k=20$.
- Let $m=11$
- Choose, $x=mk+1$. Thus, $x=222$. But correspondingly, there is no solution for y .
- So, we continue until $x=mk+4$. $x=224$. Thus, $y=248$ and hence $m=11$ is encoded as $(224,248)$.
- Decoding: $m=\text{floor}((224-1)/20)=11$



Therefore, m is encoded as x comma Y , and the decoding is very simple, you take this x you subtract one from it, divide it by k , and take the floor of this value **so that is your m so x minus 1 divided by k and you take a floor of this this** will give you m . Therefore, this is the way how you can get this value of m from this pair of x comma y . So, I will give you an example on this, that is suppose p is seven hundred and fifty one, a is minus one, b is one hundred and eighty eight, and k is twenty. Let this be some arbitrarily

chosen values, and here m is again eleven, and I choose this X as x equal to m into k plus one.

(Refer Slide Time: 25:17)

$m = 11$
 $x = m \cdot (20) + i$
 $i = 1: x = (11 \times 20 + 1) \bmod p$
 $= \underline{221}$

$1 \rightarrow 19$
 $y^2 = (x^2 + ax + b) \bmod p$
 $20 \times 11 + 1$

(Refer Slide Time: 25:21)

Example

- $p=751, a=-1, b=188$ and $k=20$.
- Let $m=11$
- Choose, $x=mk+1$. Thus, $x=222$. But correspondingly, there is no solution for y .
- So, we continue until $x=mk+4$. $x=224$. Thus, $y=248$ and hence $m=11$ is encoded as $(224,248)$.
- Decoding: $m=\text{floor}((224-1)/20)=11$

So here if I take this value m being eleven and k being twenty, if you multiply eleven into twenty and add one you get two hundred and twenty two is it correct, I think is a mistake, it is twenty into eleven plus one that is two hundred and twenty one. Therefore, please correct it this is two hundred and twenty one, so but correspondingly we can see there is no solution for y. If you take this value of X and plug in to the curve equation,

then you will not get an integral value of y . So that you can check again, but **this we can** this you can check offline. Similarly, we continue like this and you keep on incrementing the value of this one, two, three and so on and you will find in this case that when this i is equal to four your x b equal to two hundred and twenty four, and then you will get there is a corresponding value y equal to two hundred and forty eight, which is an integral solution. So that means that this m equal to eleven can be encoded as 224 comma 248.

So, now I want to obtain back the value of m , from this I take this two hundred and twenty four subtract out of 1 and divide it by twenty, and then take the seal of this when that is equal to eleven. So that is essentially how you get back m from the pair x comma y . Now, I think this should be (()) to us **that is y** there will be at least one value for which **this y is I mean for which this pair is a value pair** so I am leaving that you to think.

But these are very simple encoding technique, there could be other principles or other methods which you can do the encoding also. What is the probability that you will get at least one such thing, I mean if **you try keep on try** it will be so on you want to try k number of times. **there at least one of them will be possible encoding** that means the probability is one by k , therefore we repeat this k number of times. You should get one solution and the decoding is simple, decoding exactly gives you back the starting message m .

(Refer Slide Time: 27:08)

**Example – Elliptic Curve
Cryptosystem Analog to El Gamal**

- Alice chooses another random integer, k from the interval $[1, p-1]$
- The ciphertext is a pair of points
 - $P_c = [(kB), (P_m + kP_B)]$

NPTEL

Therefore, we have solved this, that is we have done and obtain this value of P M. So, this P M is the encoded thing, that is we have taken the message M, encoded it on to the elliptic curve it becomes P M. So P M is again a point, that is it is a pair of x comma y may that it is a is that x comma y it is Cartesian point. So, now I want to apply and generate the cipher text, so in order to generate the cipher text I have to use the public key. So what I do or what Alice does is that, you choose a random integer k from the integer one to P Minus one therefore it choose the random integer from 1 to P Minus 1 call it k and multiplies it by Bob's public key. So, it takes Bob's public key as P B and multiplies it with k, it multiplies k with P B and adds it with P M, note all these additions are defined on the curve, therefore you chose this, and the other pair is k into B. You already had the base point B, you multiply it scalarly by the corresponding k which you have chosen, and you are passing this pair as a cipher text.

So this reminds us of the El Gamal encryption, I mean what we have seen previously this is exactly similar to that. Therefore, it is called an elliptic curve cryptosystem which is analogous to the El Gamal cryptosystem. So, you choose P M and you add it to k into P B, P B was your public key of Bob, here multiplied with k and passed it.

(Refer Slide Time: 29:09)

© CET
I.I.T. KGP

$$m = 11$$

$$x = m \cdot (20) + i$$

$i \rightarrow 19$

$$i = 1: x = (11 \times 20 + 1) \bmod p$$

$$= \underline{\underline{221}}$$

$$y^2 = (x^3 + ax + b) \bmod p$$

$20 \times 11 + 1$

$$\frac{P_M + kP_B}{kP_B} = \frac{P_M + k(b+B)}{kP_B}$$

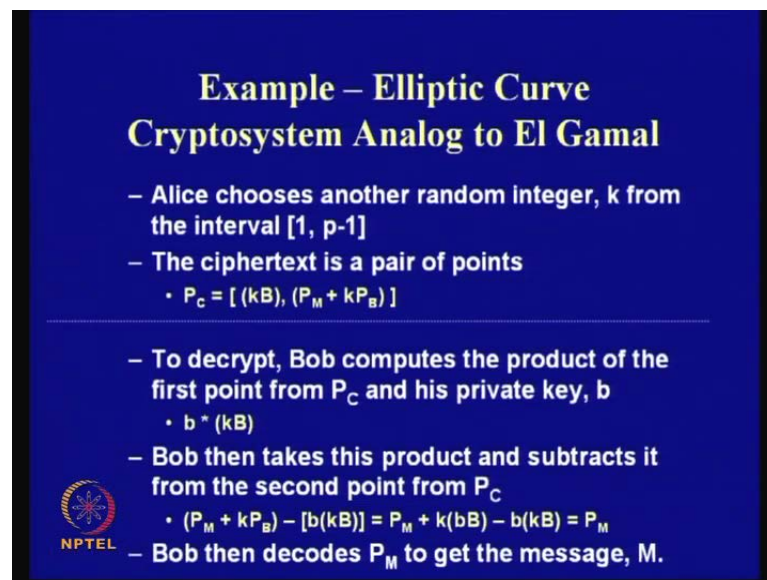
} Bob receives the cipher-text.
Bob has b.
 $b(kB) = k(bB)$
 $= \underline{\underline{kP_B}}$

NIPTEL

So, now you can note that in this particular exponent, if you want to obtain P M from this P M plus K into P B, then what do you need to obtain. So the cipher text has got P M plus K into P B, and what is this, this equal to P M plus K into the secret b and multiplied


by the base point B . What was the other corresponding component there, the other component was kB , so then Bob receives the cipher text and Bob has b , because b is the corresponding private key. So what Bob does is that, Bob takes the first component kB and multiplies kB with the corresponding value of b that is, if you multiply this kB with b then what you obtain is nothing but k into b into B , and that is nothing but k into P_B . Therefore, Bob by using its own private key can actually obtain the value of k into P_B and after that it is simple, because you have to just subtract out this value so that you get value of P_M . That is what is shown in (refer slide time: 30:37).

(Refer Slide Time: 30:37)



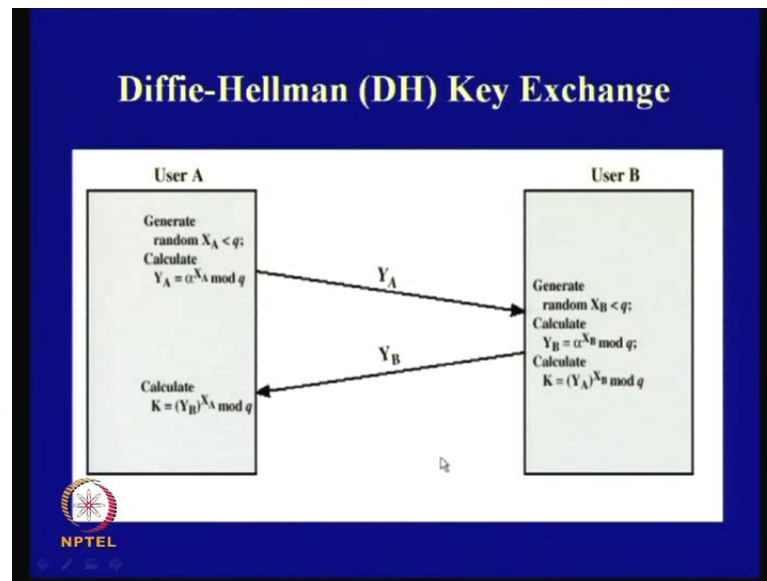
Example – Elliptic Curve Cryptosystem Analog to El Gamal

- Alice chooses another random integer, k from the interval $[1, p-1]$
- The ciphertext is a pair of points
 - $P_C = [(kB), (P_M + kP_B)]$
- To decrypt, Bob computes the product of the first point from P_C and his private key, b
 - $b * (kB)$
- Bob then takes this product and subtracts it from the second point from P_C
 - $(P_M + kP_B) - [b(kB)] = P_M + k(bB) - b(kB) = P_M$
- Bob then decodes P_M to get the message, M .

 NPTEL

That is, if you want to do the decryption then you take this b and you apply it over kB and then you just do this subtraction operation, it will eliminate these two terms to obtain back the value of P_M . So, P_M is the encoded message on the elliptic curve, then again by using the previous decryption or the decoding algorithm, rather you can actually obtain back the corresponding value of m , the message. This is quite analogous to what we have seen. So, Bob then decodes P_M to get the message m , therefore you have to do the final decoding to get back the original message. This is how you are actually doing the operation of encryption and decryption.

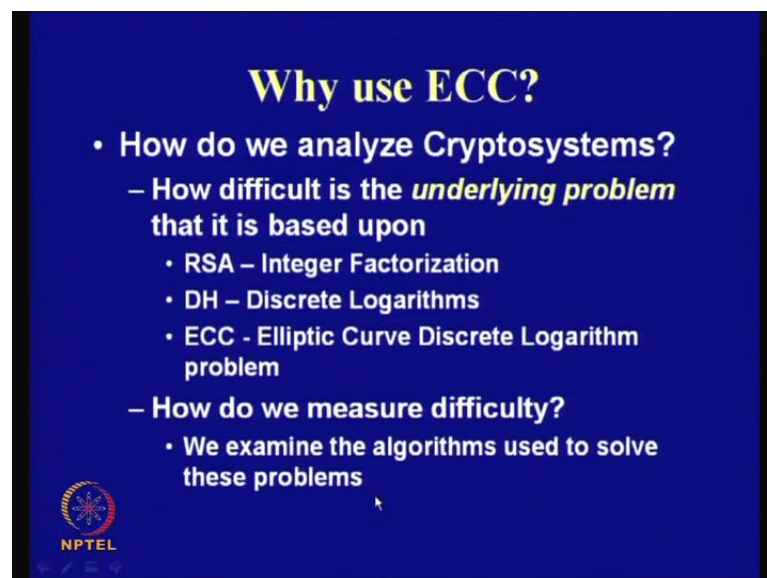
(Refer Slide Time: 31:21)



So, we are obviously extend it for signatures, the authentication that we are not actually going to and that is quite straight forward, rather the other interesting application of elliptic curves is in the Diffie-Hellman key exchange. If we, just to recap that is what we have seen in context to the finite field p , **there is a** there is value of alpha, you choose alpha power of X_A modulo q and obtain back Y_A , and send this Y_A to the corresponding user B. So what the user B does is that, the user computes Y_B by again choosing alpha and raising it to some secret value call it X_B , and sending it back to the user A. So what both user A and user B does subsequently is that, they uses its own secret value and raises Y_B to that power. What B does is that, it takes Y_A which is obtained from user A and raises to X_B , and what we know is that this value and the output value is same. Therefore, that is a final shared or exchanged key which is used for subsequent encryptions.

So details is like this, that is you choose a private key a for example, Alice has a private key a and a public key as P_A , and similarly Bob has a private key b and a public key as P_B . So what Alice does is that, Alice obtains these corresponding value of b into B and multiplies it with its own private key a . So that is a into b into B , and that is same as what Bob computes, because Bob obtains a into b from Alice and that it raises it or multiplies it with b , and since a into b is same as b into a both these two values are same. So, this is the final shared key of a into b into B **so that is basically that** you can also apply elliptic curves to perform the original Diffie-Hellman exchange as well **this symbol right so this is same I mean** what we have seen previously.

(Refer Slide Time: 34:22)



Why use ECC?

- How do we analyze Cryptosystems?
 - How difficult is the *underlying problem* that it is based upon
 - RSA – Integer Factorization
 - DH – Discrete Logarithms
 - ECC - Elliptic Curve Discrete Logarithm problem
 - How do we measure difficulty?
 - We examine the algorithms used to solve these problems

NPTEL


So now, why do we use ECCs? To understand that, how do we analyze cryptosystems. So if I want to say that whether RSA is greater than ECC, then we have to compare the difficulties of their underlying problems. For example, RSA is based upon the integer factorization, your Diffie-Helman is based on some problems which is discrete; I mean is actually based on Diffie-Helman assumption, and similarly if you want to see the El Gamal cryptosystem that is based on your discrete logarithm problem, and elliptic curve cryptography is again based on elliptic curve discrete logarithm problem. Therefore, how do we measure the difficulty, we examine the algorithms which are used to solve these problems.

(Refer Slide Time: 35:14)

Security of ECC

- **ECC results in shorter key sizes.**
 - this leads to efficient applications, at the same level of security.
 - e.g 163 bits of ECC key is equivalent to 1024 bits of a RSA key.

ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15 360	1 : 30	256




For example, here are some values which shows that elliptic curve cryptosystem results in shorter key sizes. Therefore, if you compare for example like 163 bits, you will see that a 1024 bit RSA key size is equivalent to that of a 163 bit elliptic curve key size. So, here you see that there is a significant shortening of the key size, the key size is significantly short it is almost one is to six ratio, and as we are going down the ratio is actually increasing. So which means that if you want to apply or rather develop public key cryptosystems on resource constant environments, then elliptic curve is actually a more promising public key cipher.

(Refer Slide Time: 36:01)

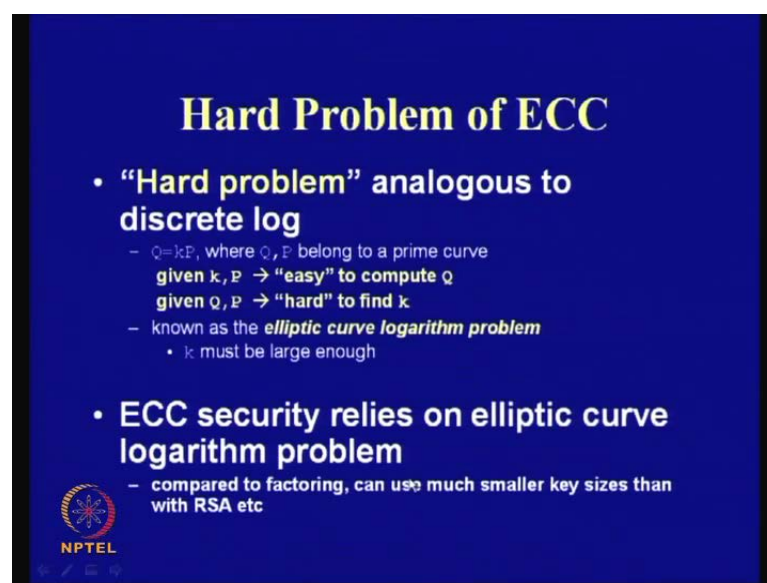
Applications of ECC

- **Many devices are small and have limited storage and computational power.**
- **Where can we apply ECC?**
 - Wireless communication devices
 - **Smart cards**
 - **Online Transactions**
 - **Web servers**
 - *Any application where security is needed but lacks the power, storage and computational power that is necessary for our present day applications.*




Therefore, you will find that if the devices are small they have limited storage, and they have computational power restrictions where actually elliptic curve cryptography is ideal. It is actually trying [v] program is being adopted worldwide, therefore if you want to apply for wireless communications, for smartcards, develop online transactions web servers that is any application where security is needed. But also the place of platform lacks sufficient power storage and computational requirement, therefore this is very motivating and very useful for the present day applications.

(Refer Slide Time: 36:39)



Hard Problem of ECC

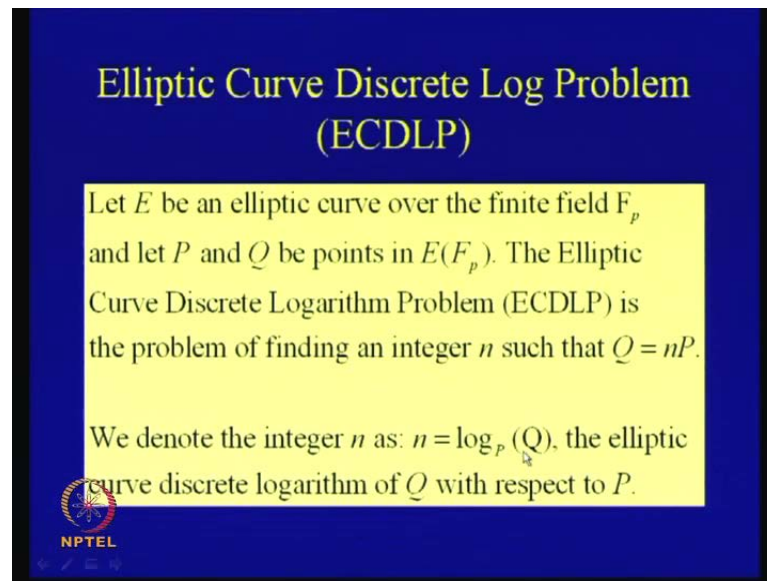
- “Hard problem” analogous to discrete log
 - $Q = kP$, where Q, P belong to a prime curve
 - given $k, P \rightarrow$ “easy” to compute Q
 - given $Q, P \rightarrow$ “hard” to find k
 - known as the *elliptic curve logarithm problem*
 - k must be large enough
- ECC security relies on elliptic curve logarithm problem
 - compared to factoring, can use much smaller key sizes than with RSA etc

 NPTEL

So, the question is why essentially RSA has got such a bigger key size and elliptic curve cryptography has a comparatively much smaller key size. In order to answer this, it has to what we do with the underlying hardness of the problem. That is, what is the corresponding difficulty of solving the inherent internal problem that is how hard is it to solve the Diffie-Hellman problem in elliptic curves, and how hard it is to solve the Diffie-Hellman problems in F_p , so that is the basic question.

So your elliptic curve security therefore the question is that, the hard problem is analogous to discrete logs that is Q is equal to k into P . So the question is that, there is a point P , and there is a point Q , and the question is that given k and P it should be easy to compute Q , but it is hard to find the value of small k that is the scalar. So this problem is known as the elliptic curve discrete logarithm problem.


(Refer Slide Time: 38:07)



Elliptic Curve Discrete Log Problem (ECDLP)

Let E be an elliptic curve over the finite field F_p and let P and Q be points in $E(F_p)$. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is the problem of finding an integer n such that $Q = nP$.

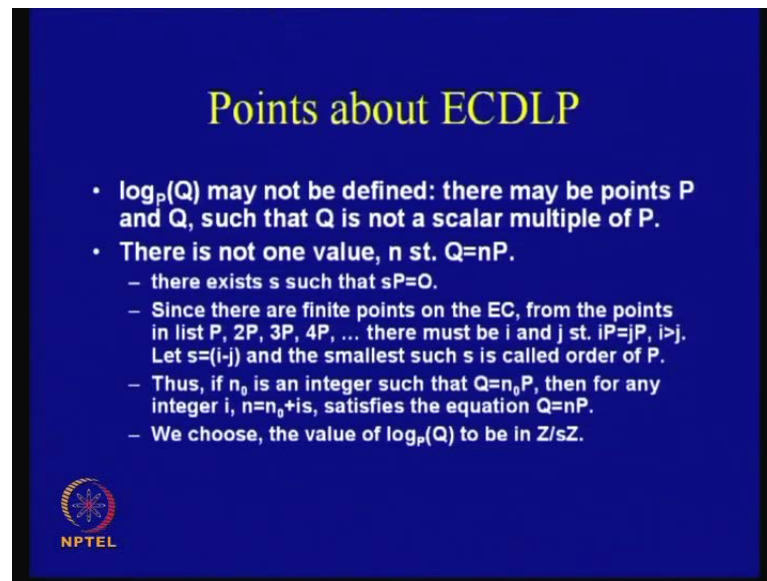
We denote the integer n as: $n = \log_P(Q)$, the elliptic curve discrete logarithm of Q with respect to P .

 NPTEL

So in this case, we know that k must be large enough that is, if k is small say one or two obviously it is not difficult but it becomes difficult when this k is really large. So elliptic curve security relies on elliptic curve logarithm problem and it can be compare to other problems that we know. So these are the basic definition of elliptic curve discrete logarithm problem more formally.


So let E be an elliptic curve over the finite field F_p , and let P and Q be points in this $E(F_p)$. Therefore, there is an elliptic curve and there are two chosen points P and Q , and we are actually interested in the elliptic curve discrete logarithm problem. So, the elliptic curve discrete logarithmic problem is the problem of finding an integer n such that Q equal to nP is satisfied that is, given this value of P and given this value of Q , how hard it is to compute this value of integer n . So we denote this integer n as n is equal to $\log_P(Q)$ base P , so that is the elliptic curve discrete logarithm of Q with respect to P . **that is the elliptic curve** I mean these are the elliptic curve discrete logarithm of Q with respect to P .

(Refer Slide Time: 39:18)



Points about ECDLP

- $\log_p(Q)$ may not be defined: there may be points P and Q , such that Q is not a scalar multiple of P .
- There is not one value, n st. $Q=nP$.
 - there exists s such that $sP=O$.
 - Since there are finite points on the EC, from the points in list $P, 2P, 3P, 4P, \dots$ there must be i and j st. $iP=jP, i>j$. Let $s=(i-j)$ and the smallest such s is called order of P .
 - Thus, if n_0 is an integer such that $Q=n_0P$, then for any integer $i, n=n_0+is$, satisfies the equation $Q=nP$.
 - We choose, the value of $\log_p(Q)$ to be in $\mathbb{Z}/s\mathbb{Z}$.


NPTEL

So now, we will see certain interesting properties of this elliptic curve discrete logarithm problem. One of the thing is that, this $\log Q$ base P may not be defined that is, you may choose two points Q and P such that this $\log Q$ base P is actually not defined, because it is not necessary that always you will find that $Q = nP$ maintains the relation that is q is equal to n into p . So it may be that, there are two points for which Q is not equal to n into P for any integer n . That is actually something which is to be kept in mind, but the thing is that we actually are not really faced with any problem because of the nature in which we are actually applying our elliptic curve cryptography. Whenever we are applying, elliptic curve cryptography is by repeated additions.

So the corresponding Q and P that we are generating or that we are concerned of actually has got this relation, but it is not true for any arbitrarily chosen Q and P values. Therefore, **since there are fine I mean therefore** it is important to be kept in mind that, it is not necessary that for any Q and P this elliptic curve or this discrete logarithm has to be defined, but **none other** fashion in which we are applying this elliptic curve we do not face any significant problem.

The other important thing is that, there is not a single value, for which this holds. That is it is not true that this $Q = nP$ is true only for one unique value of n , there can be multiple solutions to this equation. So let us just try to see this because this is quite straightforward, that is suppose you choose any point P there will exist an s such that s

into P will be equal to O . So that follows from the basic property of the underlying group that is, if you keep on adding P then there will exist one such integer s for which s into P is equal to the identity of that O , and the least such value for which this holds is actually called the order of the point P .

So, now the other fact that is also important is that, you know that the points on the elliptic curves are actually finite set of points, so if you just take any point P and you consider P $2P$ $3P$ $4P$ and so on, there must be two such values i and j for which they are the same because it is a finite set of points. Therefore, if you choose i there must exist some value j , there must exist some value of j where i is greater than j , say without any loss of generality for which i into P is equal to j into P . That means if you rearrange, then i minus j into P is equal to O . So if you say that s is equal to i minus j and the smallest of s is called the order of P , so there we know that there must exist some case for which this i into P is equal to j into P . Therefore, if n_0 is an integer such that Q is equal to $n_0 P$, there is one such integer n_0 for which Q equal to $n_0 P$ is satisfied.

(Refer Slide Time: 43:22)

© CET
I.I.T. KGP

$$\begin{aligned}
 Q &= nP \\
 &= (n_0 + iS)P \\
 &= n_0P + i(SP) \\
 &= n_0P
 \end{aligned}$$

$\text{eg } O \rightarrow n$
 $(P_1, P_2) \rightarrow \mathbb{Z} \setminus s\mathbb{Z}$

RIPTEL

So you can actually choose any n by adding n_0 with i multiplied by s , then this also will satisfy this equation of Q equal to $n P$, because you know that Q equal to $n P$ is nothing but here n_0 plus i integer, the order s that is the order of the point P , and you are multiplying with P . Therefore, that is equal to n_0 into P plus i into $s P$, and what is $s P$, s

P is O . So what you get back is as $n \cdot 0 \cdot P$, and you know that Q equal to $n \cdot 0 \cdot P$ is satisfied, therefore this actually holds.


So this logarithm is, actually when you are considering this logarithm of the point Q with respect to point P , you are obtaining a corresponding integer value, it is actually a mapping. Mapping from what to what, mapping from two points, Q is a point, P is a point. So there are two points on the curve, say I call it a point P_1 and point P_2 **there are two points on the curve** to an integer n , and this integer you can actually denote as Z and you can actually choose it as $s \cdot Z$, because any such thing will satisfy that is any $n \cdot 0$ plus i into s will satisfy. But we are actually interested in the least representative of this class.

Therefore, what we choose as the least difference of the class is either by this or essentially this is my minimum set, therefore I can actually obtain various classes and I am interested in this congruence class. Therefore, that is the way how we choose this corresponding thing, and we are interested in the least such value. We are interested in actually **sorry I mean this probably a more customize thing $z \cdot s \cdot z$** . So do you understand this that is, essentially there can be more than one solutions to this discrete log problem, but what we are interested is a least such value. Therefore, if we know the order of the point P is s , we can actually take any value which satisfies this, and we can actually obtain the corresponding modulus with respect to s which should also satisfy this equation. So there are large numbers of solutions, you can actually keep on generating solutions like this.

(Refer Slide Time: 45:57)

How hard is ECDLP?

- Consider 2 lists, generated by choosing random integers j_1, \dots, j_r and k_1, \dots, k_r between 1 and p .
 - List $L_1: j_1P, j_2P, \dots, j_rP$
 - List $L_2: k_1P+Q, k_2P+Q, \dots, k_rP+Q$
 - Any collision between the 2 lists imply, $j_uP = k_vP+Q$, thus $Q = (j_u - k_v)P$.
 - From B. Paradox, with $r = O(p^{1/2})$ there is a good chance of a collision.
 - The fastest algorithm to solve ECDLP is $O(p^{1/2})$.
 - The ECDLP is harder than the DLP in F_p^*
 - The DLP problem has faster algorithms.



 NPTEL

Now the question is how hard is the elliptic curve discrete log problem? So we have studied in context to the previous algorithms like, in context to previous RSA when we have studied El Gamal cryptosystem, we actually need not go much into the crypt analysis of El Gamal cryptosystems, but they are actually algorithms which are better. For example, if you remember in context to factorization, we have discussed about algorithm which had a complexity of $O(n^{1/4})$, because if you want to generate, if you remember that algorithms like Pollard's rho algorithm are quite efficient algorithms in order to factorize or rather to find the prime factors of a given composite number. So if that problem is solved then RSA problem is also solved.

Now the question is how hard is the elliptic curve discrete logarithm problem? For example, just if you remember the birthday paradox, consider an approach which is similar to the birthday paradox. So if I generate two lists L_1 and L_2 based upon say r random choice the values like j_1 to j_r and k_1 to k_r , so what I do is that I just generate j_1 to j_r randomly, and all these things are actually from one and P ; between one and p you choose any arbitrary values.

$i = (j_1 - p) + j_2 - p$ and so on till $j_r - p + i$ generated. Similarly, another list which is built upon like, k_1 into P plus Q , k_2 into P plus Q and then k_r into P plus Q . So if you find that, in this two list there is a collision, so $j_u - P$ that is for some u is equal to $k_v - P + Q$. Therefore, we can immediately rearrange or one solution is $j_u - k_v + P$, that means if I get this list and there are two such values for which the common term occurs, then you can actually solve the elliptic curve discrete log problem.

Now, how difficult or rather what should be the least value of this value of r for which you get a collision. So actually you know that these values being from one to p there are P such values, and if you apply the birthday paradox, and if you set your r to be of the order of P to the power of half or square root of P you know that we have got a very good chance or very high probability of finding a collision. That is the customary birthday paradox approach. So, if r is the order of the P power of half, then we can actually obtain an elliptic curve discrete log solution. This is probably the kind of fastest algorithm that has yet been found to solve a elliptic curve discrete log problem which has got a complexity of $O(\sqrt{P})$.

So there are algorithms of the nature of humbling which are called as index calculus algorithms which actually have the runtime of the order of P power of half. Whereas, for previous things like, for the discrete log problems in F_p there are more advanced and fastest algorithms. So you may remember that you can quite easily find out whether I mean I mean not easily rather but you can find if you want to factorize large composite number. And if you apply a polar slope kind of algorithm, then you remember that the complexity of that was if the factor of that if n can be factored as P into Q , then the complexity of that was $O(P \text{ power of } \sqrt{P})$, and P is of the order of \sqrt{n} . Therefore, the complexity was n to the power of $1/4$, so that was a faster kind of algorithm.

So in the context of this elliptic curve discrete log problem, the fastest algorithm which has been found to solve this elliptic curve discrete log problem is of the order of P power of half, and there are actually developments of course in the context of elliptic curve discrete log problem by bringing in a concepts of pairings. So first of all, pairings in cryptography, we are first actually used to solve the elliptic curve discrete log problem that is they are used in that sense, in a (()) sense. But after that, it was actually used to a positive sense in developing ciphers, but the first use or usage of pairings was in the cryptanalysis of elliptic curve discrete log problem. So the general comment that we can make is that, the elliptic curve discrete log problem is probably harder than the discrete log problem in F_p . The discrete log problem in F_p has actually got faster algorithms that means actually in elliptic curve discrete log problems we can do with a shorter key size. That is the reason why we can actually do with shorter key sizes.


(Refer Slide Time: 51:04)

Points to Ponder

- Suppose $p > 3$ is an odd prime, and $a, b \in \mathbb{Z}_p$. Further, suppose that the equation, $x^2 + ax + b = 0 \pmod{p}$ has 3 distinct roots in \mathbb{Z}_p . Prove that the corresponding group $G = (E, +)$ is not cyclic.

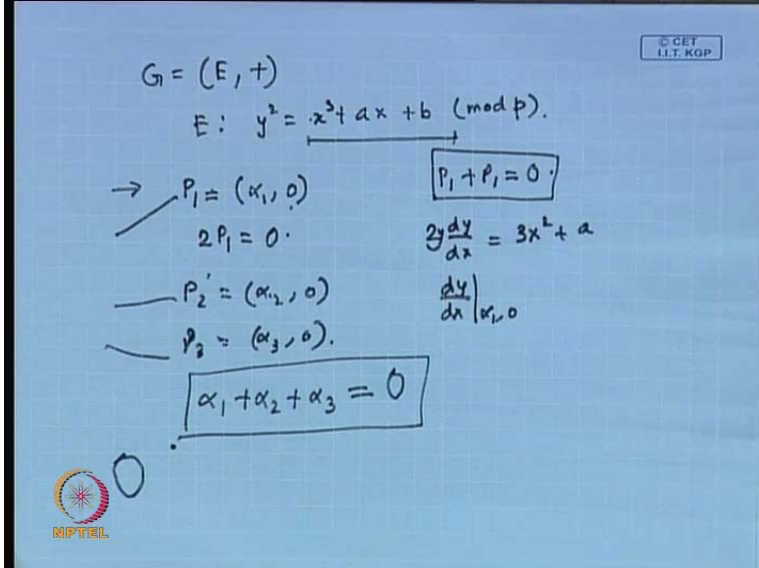
– Hint:

- Prove that $P_1 = (\alpha_1, 0)$ has order 2.
- Show that $P_1 = (\alpha_1, 0)$, $P_2 = (\alpha_2, 0)$ and $P_3 = (\alpha_3, 0)$ (where α_1, α_2 and α_3 are the 3 distinct roots) along with the point O are isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.
- Show that the subgroup is not cyclic.



So I can give one question here, that is which you can take and think upon is that, suppose there is P which is greater than 3, these are odd prime and a and b both belongs to \mathbb{Z}_p . So the equation of this will be $x^3 + ax + b = 0 \pmod{p}$. So one question is that, prove that a corresponding group that is G equal to E plus is actually not a cyclic group. **it is not a cyclic group**

(Refer Slide Time: 52:00)



$G = (E, +)$
 $E: y^2 = x^3 + ax + b \pmod{p}$


$\rightarrow P_1 = (\alpha_1, 0)$
 $2P_1 = O$

$P_2 = (\alpha_2, 0)$
 $P_3 = (\alpha_3, 0)$

$\alpha_1 + \alpha_2 + \alpha_3 = 0$

$P_1 + P_1 = O$

$2y \frac{dy}{dx} = 3x^2 + a$
 $\frac{dy}{dx} \Big|_{\alpha_1, 0}$



So therefore, you remember the definition of cyclic group that is, in this case you can actually take any generator, and you can immediately apply the addition operation on

that, and you should be able to construct the entire proof. The corresponding curve **corresponding** or rather corresponding group is G equal to E plus, where E is defined as y^2 equal to x^3 plus a x plus b , and you are doing a modulo p operation. So the question is that, you have to prove that corresponding group here is actually not a cyclic group. We have given you some hints here, so first thing is to prove that, if you take a point P_1 call it α_1 comma 0 , this has actually an order of two. Order of two means that if I take P_1 , and if I add P_1 with P_1 I should get O . So this is the first exercise which you can take that is if I take P_1 , add P_1 with it then I get back O .

So for that it is very simple, if I want to obtain the doubling operation here, then I have to take a differentiation of this. So that is a small catch here, that is you cannot apply the doubling equation straight forward, because in doubling if you remember in, the radiation we had assumed is that y was not equal to zero. But here we have y equal to zero of this point, so if I take a differentiation here $2y \frac{dy}{dx}$ it is $3x^2$ plus a that is a point therefore, the $\frac{dy}{dx}$ at the point of α_1 comma 0 is not defined because you actually dividing by $2y$.

So that is a vertical line, therefore the vertical line will intersect in elliptic curve and the point O . therefore, two into P_1 is actually equal to zero, that is why the order is two. Similarly, you can actually derived points like P_2 which is α_2 comma 0 , and P_3 as α_3 comma 0 , all of them have got order of two. Now, if this α_1 , α_2 and α_3 are distinct roots of these three equations or rather these equation, then you know that α_1 plus α_2 plus α_3 is actually equal to zero because there is no term which is an x^2 term.

Therefore, α_1 plus α_2 plus α_3 is equal to zero. So using this you can actually show that this P_1 , P_2 and P_3 along with the point at infinity O actually forms a group, that is it forms a sub group. And the other point to be noted here is that, this subgroup that is consisting of P_1 , P_2 , P_3 and O , so sub group means first of all we have to define that it is an closed set.


(Refer Slide Time: 55:20)

Points to Ponder

- Suppose $p > 3$ is an odd prime, and $a, b \in \mathbb{Z}_p$. Further, suppose that the equation, $x^2 + ax + b = 0 \pmod{p}$ has 3 distinct roots in \mathbb{Z}_p . Prove that the corresponding group $G = (E, +)$ is not cyclic.

– Hint:

- Prove that $P_1 = (\alpha_1, 0)$ has order 2.
- Show that $P_1 = (\alpha_1, 0)$, $P_2 = (\alpha_2, 0)$ and $P_3 = (\alpha_3, 0)$ (where α_1, α_2 and α_3 are the 3 distinct roots) along with the point O are isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.
- Show that the subgroup is not cyclic.



The other thing is that, this sub group is actually not a cyclic subgroup. Here there is no generator which will generate all of them, and that is quite simple to check. And if the sub group is not cyclic, then the group is also not cyclic. So this is actually a problem from Stinson, therefore this is the sketch of the solution which you can complete.

So these are some of the references along with I am also using the standard Douglas Stinson's book as reference. So next day's topic will be implementation of elliptic curve cryptography.