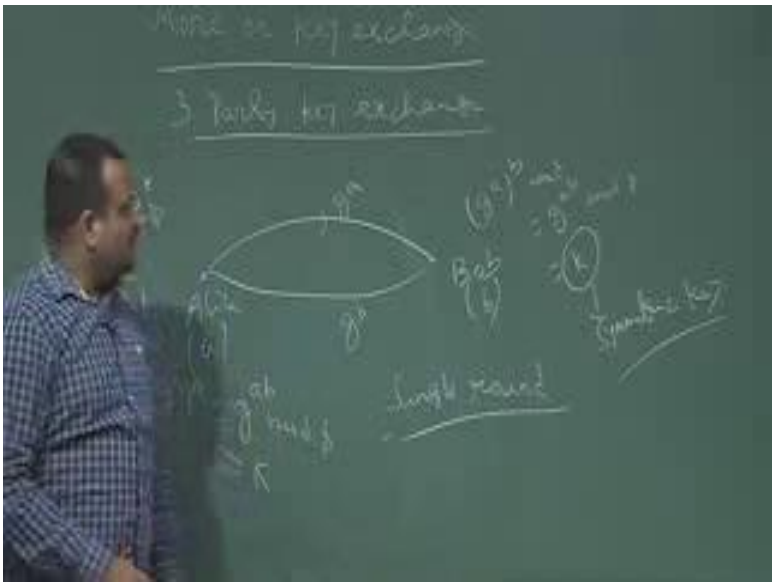
**Internetwork Security**
**Prof. Sourav Mukhopadhyay**
**Department of Mathematics**
**Indian Institute of Technology, Kharagpur**

**Lecture - 45**
**More on Key Exchange Protocol**

So, we will talk about more on key exchange protocol like typically three party key exchange protocol. So, so far we have seen if there are two party, how we can do the key exchange. So, we have the Diffie-Hellman key exchange protocol for two party we have seen that, so Alice and Bob.

(Refer Slide Time: 00:35)



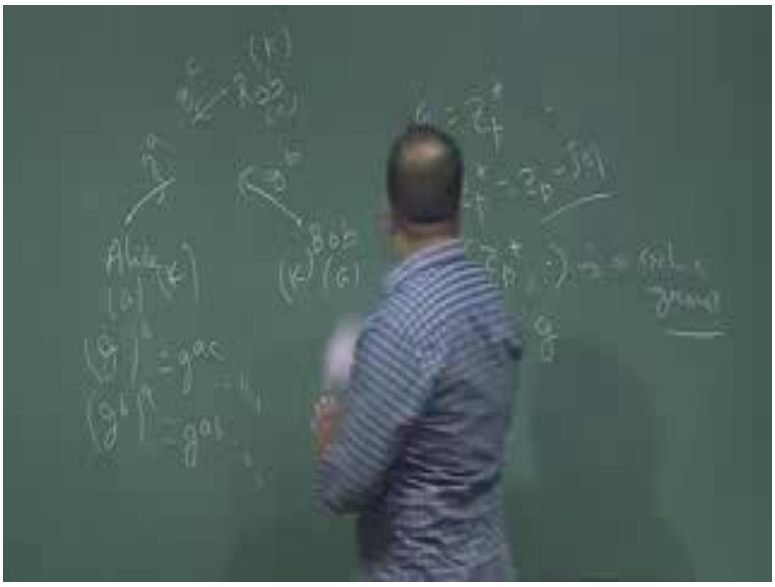
So, now we will talk about three party, if we have three party three parties then key exchange protocol. So, how three party can exchange? If we have two party like Alice and Bob, now we know that they can just by Diffie-Hellman they can in a single round they can do the key exchange like Alice can choose. So, they have a group G. So, this is typically Z p star, it could be other group also. So, where p is a prime then and g is a generator of this group small g.

So, Alice compute g to the power a, Bob is choosing this b. So, Bob compute g to the power b. So, then Bob is computing g to the power a to the power b mod p, so which is basically g to the power a b mod p which is the key, and Bob is receiving this g to the power b and Bob is having g to the power a. So, Bob is computing g to the power a, so g

to the power a b mod p. So, this is basically same key. So, they agree with the common key in a single round, this is single round protocol. So, single round, single round key exchange protocol

So, just one communication is required to establish this key. Now, they can use the secret key this is the symmetric key. So, then now they can use this symmetric key for their further communication encryption, decryption.

(Refer Slide Time: 02:48)



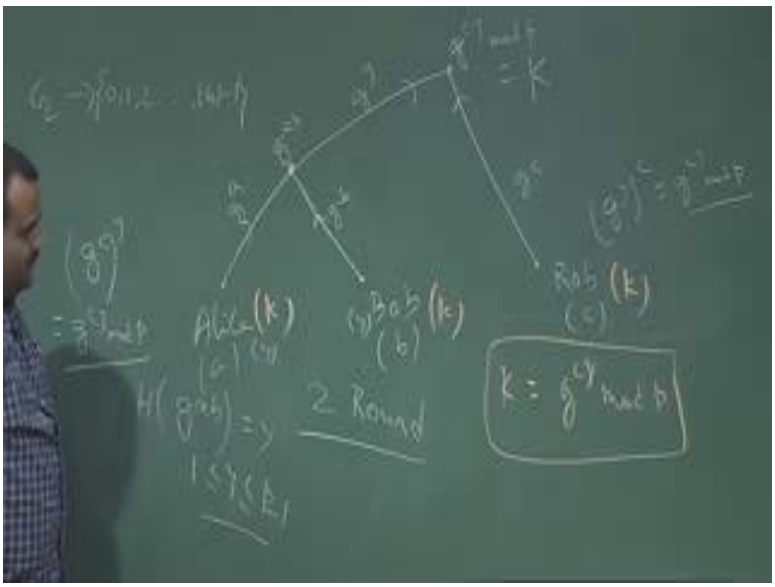
Now, this is two rounds, now we want to extend this or we want to see this how we can do the key exchange, if there are three parties. So, so we have if we have three party say Alice, Bob and we have another party say Rob. Now the question is how they can agree with a common key. So, key exchange between three parties. So, we want to see whether we can extend the Diffie-Hellman protocol for this three party exchange key protocol. So, if they are choosing this a, b, c, so they are under Z, so g be a group Z p star say it could be general group also, but for simplicity we are taking Z p star; and Z p star is a group under multiplication p is a prime. So, Z p star is basically Z p minus zero, we have to remove the additive identity element then these will form a cyclic group. So, if it is cyclic group means it has a generator, and suppose g is a generator of this group.

Now how they can agree with a common key? So, basically we need to do like this. So, this cannot be done a one single round, because if Alice is sending g to the power a, so say Rob if Alice say Rob is sending g to the power c Alice is sending g to the power a

and Bob is sending g to the power b, then how they can agree. So, they can mutually agree with a common key like these two party can agree with a k 1 which is basically, but we want they should agree with a single key.

So, like under this, so what Rob can do. So, Rob can agree with a g to the power a g to the power a c with, so basically there are there will be two keys like g to the power a c will be agreed with Alice for each party, there will be two keys. So, say for Alice, Alice will receiving g to the power c, so Alice will compute g to the power c to the power a. So, this is g to the power a c; obviously, mod p. And Alice is receiving g to the power b from Bob, so g to the power b to the power a, so g to the power a b. So, basically this is k 1; this is k 2. So, each party is receiving pair of key to communicate with the other party so, but that we do not want we want these three party should agree with a single key symmetric key k. So, how we can achieve that, so that we have to talk about, but that is not possible in a one-round.

(Refer Slide Time: 06:13)




So, this is a two round protocol. So, let us we are using Diffie-Hellman. So, choosing b, so this is sending g to the power a and sending g to the power b, so they are computing g to the power a b. So, this is the key they are computing. So, this is Rob, Rob is having g to the power c. Now, they are agreed with a common key g to the power a b mod p. So, now, this is with both the party g to the power a b, this is the k, I mean g to the power a b is now with both the party. Now, what they can do? They can apply hash function on it,

and this is basically y. So, y is we want a hash function such that it is taking this group element g from to the order of the group, so 0, 1. So, we want to have a index on this. So, that we can use again for g to the power y and send it to over the public channel.

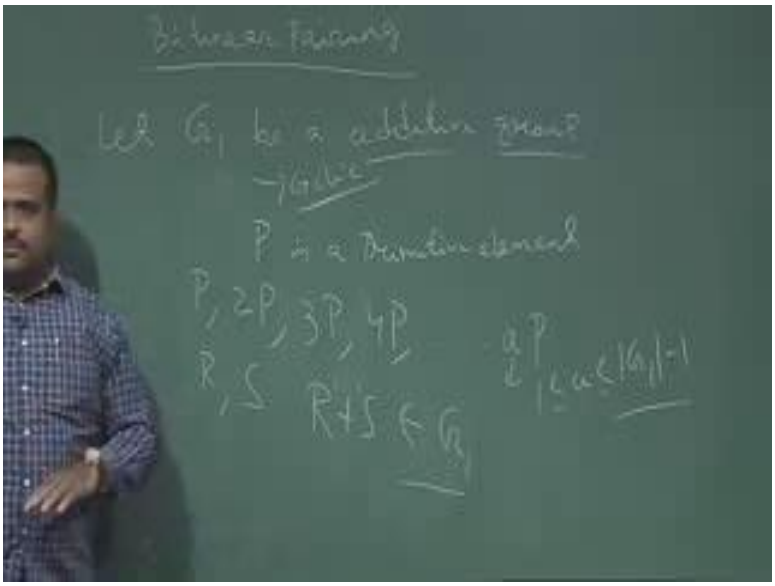
So, we want a hash function to be G to so may be 0, 1, 2 up to order of the group I mean here we are dealing with p, so it is basically p minus 1. Now in general it is order of the group. So, if we apply this hash function on this, hash function is a digest function. So, if we apply the hash function on this, so it will give us y. So, y is now p minus 1. So, now we can use y as a index. So, now, this y is shared between these two party y is known to these two party. Now, we can compute G to the power y and send it to over the public channel and then Rob can compute g to the power c. So, now g to the power c is receiving Alice, Bob. So, they can compute just g to the power.

So, Alice can compute g to the power c, g to the power c is getting from Rob over public channel. So, Alice can compute g to the power c to the power y which is Bob also can compute. So, this is basically g to the power c y mod p. And Rob is getting g to the power y; Rob is having c, so Rob can compute g to the power y to the power c. So, c this is basically g to the power c y mod p. So, this is the common key shared between these two party. So, this is basically g to the power c y mod p. So, this is the key this is k. So, this k is shared between these three parties.

So, now this is k. So, k is basically g to the power c y mod p. So, this is the three party, so c y. This is the three party key exchange protocol we are using the underline Diffie-Hellman, but this is two round; first round, they agree with a common key then second round, so this is a two round protocol two round not a single round.

Now, we want to talk about how we can have a single round protocol. So, these things, this is basically this also can be attacked by the active adversary. For, to prevent that, again we have talk about the authentication, and we can send this by signing the message like this. So, anyway I am not going to do that now, because we are just talking about key exchange protocol. So, in this lecture, we are not considering the attacks. So, this is the two round. Now, we have to talk about how we can make it single round. So, for this, what we do? We want to defined a function which is called bilinear pairing, and then we will use this function on a protocol which is called JOUX protocol to have a single round key exchange.

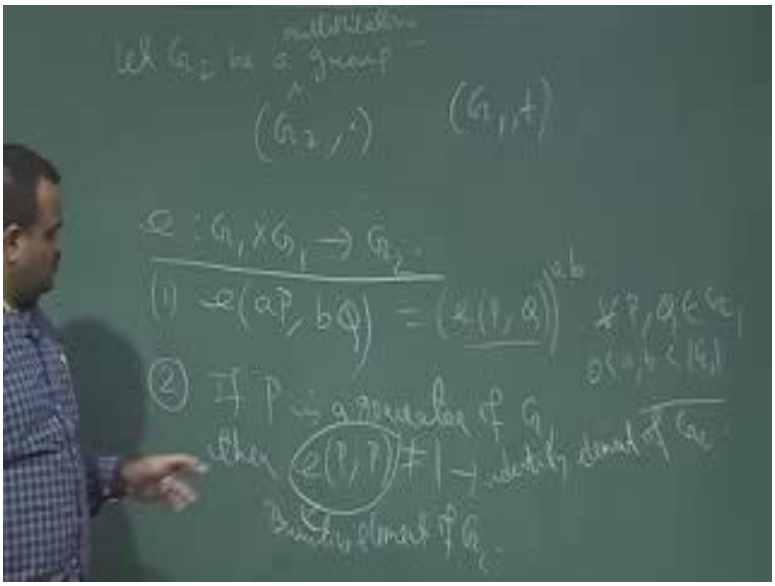
(Refer Slide Time: 11:23)




So, let us talk about bilinear pairing bilinear pairing. How we can define a bilinear pairing, it is a pairing basically bilinear pairing. So, for that let us have a group say G 1, let G 1 be a additive group and it is a cyclic group also, but it is the operation is additive sets so that means the we our operation is like if we take two element from. So, it is cyclic group, so let P be a primitive element P is a primitive element of this group. Then the elements are basically P, 2 P because it is an additive sets 3 P, like this 4 P like this. So, in general say a P, so where a is lies between 1 less than a less than equal to order of this group, a is the index, but it is additive sets it is not power. So, the operation is additive sense. So, it is just a P, 2 P like this.

So, basically if you take two element P, Q I mean here P is a primitive element. So, if you take R, S then the operation is R plus S additive sets, it is in g 1. And typically this group is a elliptic curve group, this is coming from elliptic curve points. So, you know how to have the elliptic curve points over Z p. Now, these group need to be a elliptic curve group in order to have the anyway we may talk about the example of pairing. So, this is a group G 1.

(Refer Slide Time: 14:04)



Now, we have another group G 2 which is in multiplicative sense. We have another group G 2, let G 2 be a group cyclic group be a multiplicative group; multiplicative group means operation is in multiplicative sets. So, G 1 is basically plus sets is a group and this is also a group. So, we have two group G 1 and G 2. So, now, we defined a function e which is from G 1 cross G 1 to G 2; e is a function from G 1 cross G 1 to G 2. So, this is G 1.

So, now we call this function to be a bilinear function or bilinear mapping, a bilinear pairing this function is called pairing, if it has these following properties. If we take e a P, b Q for any two element b Q from G 1 this should give us e P, Q to the power a b. And this is for all p, q from G 1; and a b are any two any two index a b, so may be 0 less than a b less than order of the group, if this is true for all such a b. So, P, Q are the element from G 1, so a p means a p means p plus p plus p a times; b Q means q plus q plus q b times. So, these are two elements in G 1. So, this is the function from G 1 plus G 2. So, then this will be an element in G 2 and this should be written as this is in multiplicative sense. So, this should be written as e P, Q to the power a b, this is one property.

And second property is degenerate property, so that means, if p is a generator of this group P is a primitive element or generator of G 1 then e P, P must be a generator of G 2 so that means, e P, P should not be 1 then e P, P is not equal to 1. What is 1? 1 is the identity element of G 2. This 1 is the identity element of G 2 so that means, if it is not the

identity element then it must generate this group. Basically, this is a primitive element of G 2, this is a generator or the primitive element primitive element of G 2, but P is a primitive element of G 1. So, this condition must be there, if this is called degenerating condition so that means, if you take a primitive element or the generator of this group then e P, P. So, P is generating this group G 1. So, e P, P must generate the group G 2, so that means, e P, P should not be 1; one is because this is multiplicative sets G 2. So, one is the identical element.

(Refer Slide Time: 18:24)


So, this is third property is that there should be an efficient algorithm to compute this pairing. We should have efficient algorithm to compute e P, Q for or e R, S because Q, we have used for the generator belongs to G 1. So, it must have efficient algorithm, efficient means there should have a polynomial time algorithm efficient in the sets polynomial time. So, it should have a polynomial time algorithm for compute this pairing. So, e is called pairing e is called bilinear pairing or just a pairing is usually defined now it is called pairing, this function e is called pairing. So, there should have algorithm polynomial time algorithm to compute this pairing. So, this is the definition of pairing.

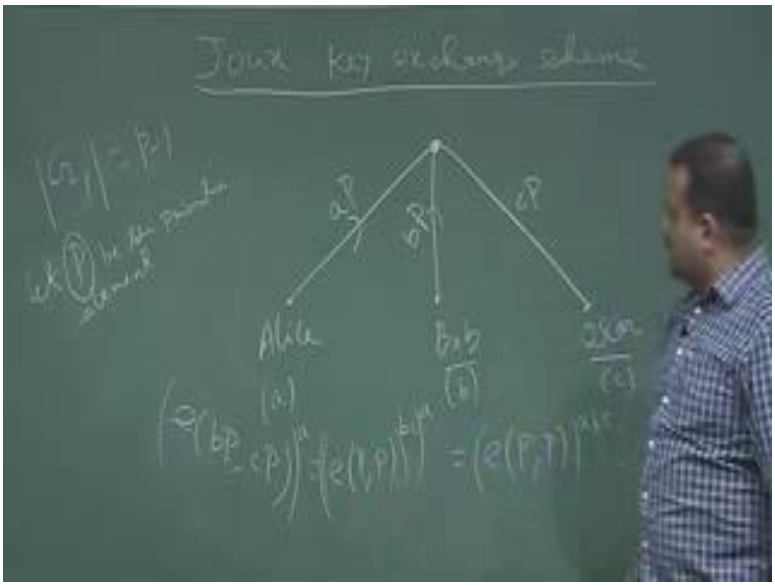
So, if a function e is having these three properties then it is called a pairing on bilinear pairing. Now, how to construct such a function that part is not so easy. So, example of a pairing it is not so easy. So, there are two example of pairing in the literature like Tate

pairing and Weill pairing, but they are all coming from so far pairings are coming from G 1; this G 1 is a elliptic curve group.

(Refer Slide Time: 20:31)


So, pairing G 1 is elliptic curve points say over Z p also then one can define the Tate pairing or Weil pairing, but the G 1 should be that is why elliptic curve part we know in additive sense. So, it is it may not be over Z p it is may be over some other group, but over Z p we can have this pairing, but this needs lots of abstract algebra concept like divisibility theorem, divisibility condition, how we can define this. So, this part we are not going to the details of this in this class particular for this course, this is out of the scope of this course. But there is a huge literature on this construction of such pairing, but pairing exist. Now, question is how we can use this for having a three party key exchange protocol in a single round, so that is called Joux protocol, Joux key exchange scheme.

(Refer Slide Time: 21:59)




So, Joux use this pairing for doing this key exchange. So, there are three parties Alice Bob and Rob. You will mind if I take Oscar which Oscar is always a bad person, but anyway maybe I can take Oscar now, he became friends of Alice and Bob which contradict the literature Oscar is always chosen as a bad guy, but anyway this is they are agreeing with a common key. So, here we are trusting Oscar. So, Oscar is now became friend of Alice and Bob. So, now, they want to agree with a common key k in a single slot single round, so how they can do it. So, they want to make use of this, what is called pairing.

So, what they do, so they know this G 1. So, G 1 is known to all G 1, G 2 anyway, G 1 is known to all. And suppose P is a generator of G 1 let p be a. So, first of all they have to choose a group G 1 on which this pairing is defined. And suppose order of this group is say p minus 1 or P and then P is a generator of this group let p be a and this is usually the elliptic curve group. So, this is usually additive group. So, we have this operation is additive sense, let p be the primitive element of this group. So, now all these parties, they choose a secret. So, now they have this G 1 is public now and this P is also public now they all choose a secret a, b, c, so where a, b, c are both coming from G 1 minus 1.

So, these are typical secret to each of this party respectively. So, a is secret to Alice which is we can say secret; b is secret to Bob c is secret to Oscar. So, a is sending to a P; Bob is sending to b P; and Oscar is sending to c P over this public channel, so that means

they are all getting this value. So, Alice Bob and Oscar this is over broadcasting. So, they are broadcasting. So, they are all getting these values, Alice is getting b P c P Bob is getting Bob is already having b P, so Bob is getting a P, c P like this. Now, how they can agree with a common key.

So, what Alice is doing? So, Alice is getting this value b P and c P. So, what Alice will do? Alice will compute the pairing e b P, c P, so if Alice will compute e b P, c P then it will be basically this is pairing. So, it is basically e P, P to the power b c. Now, Alice will compute this to the power a because a is known to Alice. So, these to the power a, so this is basically e P, P to the power a, b, c. So, this is Alice is doing, Alice is having a, now Alice is getting b P, c P over this public channel from other two party, then Alice can compute the pairing function on these two point b P, c P, and Alice can do the powering up to the power a, because a is known to Alice. So, then this is basically by properties of bilinear pairing. So, this is basically e P, P to the power b c and we have to the power a and e P, P to the power a b c. So, this is now e P, P to the power a b c is now known to Alice.

(Refer Slide Time: 27:25)


Now what Bob can do? So, Bob is having b. So, Bob is getting a P c P from other two party. So, Bob will compute e a P c P to the power b. So, this is basically e P, P to the power a P to the power a c to the power b. So, this is basically e P, P to the power a b c, because Bob is getting a P c P, so Bob will just compute this function, so that is why this

function should be computationally efficient; otherwise there is no question of computing this. So, now, Bob is also agreeing with e P, P to the power a b c. Now, what Oscar will do? So, Oscar is receiving a P b P from Alice and Bob, so Oscar will compute e a P b P to the power c, Oscar is having the secret c. So, this is basically e P, P to the power a b to the power c. So, this is basically e P, P to the power a b c. So, now Oscar is also having this e P, P to the power a b c. So, this is the secret key shared between these three parties. So, this is the k, so this is our k.

So, if k is equal to this then this k is agreed between these three party. So, this is just a one round communication. So, they are just communicating one round. So, Alice is send a P, Bob is sending b P, Oscar is sending c P and then they are by computing this they are agreed with a common key k. And now once they have the common key k, now they can securely communicate over the public channel using symmetric key encryption. So, this is a two party key exchange, but if you have three party multiparty then in current literature there are some attempt for constructing the multipiaring instead of bilinear pairing, multilinear pairing, so those can be used to, but those are under current theme of the research.

Thank you.