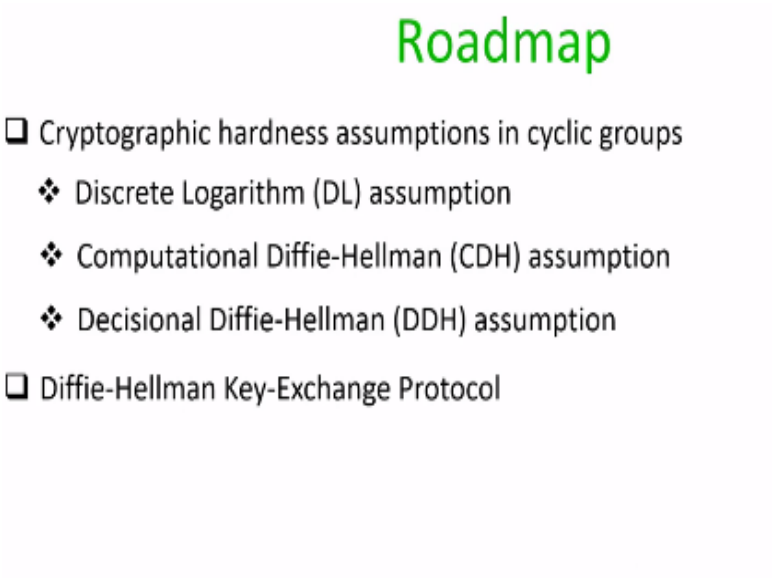
Foundations of Cryptography
Dr. Ashish Choudhury
Department of Computer Science
International Institute of Information Technology – Bangalore

Lecture - 39
Cryptographic Hardness Assumptions in the Cyclic Groups

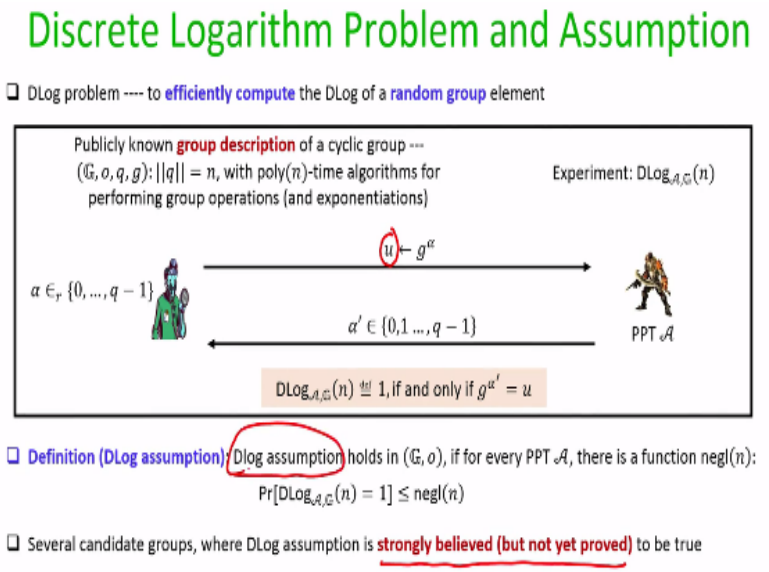
Hello everyone, welcome to this lecture. Just to recall in the last lecture we had seen the underlying ideas that are involved in the Diffie–Hellman Key Exchange Protocol. However, we had not yet seen the exact steps of the Diffie–Hellman Key Exchange Protocol.

(Refer Slide Time: 00:42)



So in this lecture we will introduce various cryptographic hardness assumptions in the context of cyclic groups namely the discrete log assumption, computational Diffie–Hellman assumption and decisional Diffie–Hellman assumption and then based on these assumptions we will see the exact steps of the Diffie–Hellman Key Exchange Protocol.

(Refer Slide Time: 01:00)



So let us first begin with the discrete log problem and the discrete log assumption. So the discrete log problem is basically that you have to compute efficiently the discrete log of a random group element where you are just given the description of the generator and a random element from the group. So this is modelled by an experiment here which played between a computationally bounded adversary and a hypothetical verifier of the experiment.

Where the description of the cyclic group is publically known and what I mean by the description of the cyclic group is that you know the underlying group operation which is there in the group and the size of the group namely the number of elements and the generator because you are given a cyclic group and we assume here that the this notation (||q||**)** double bar this notation basically means that the numbers of bits that we require to represent the value q is n.

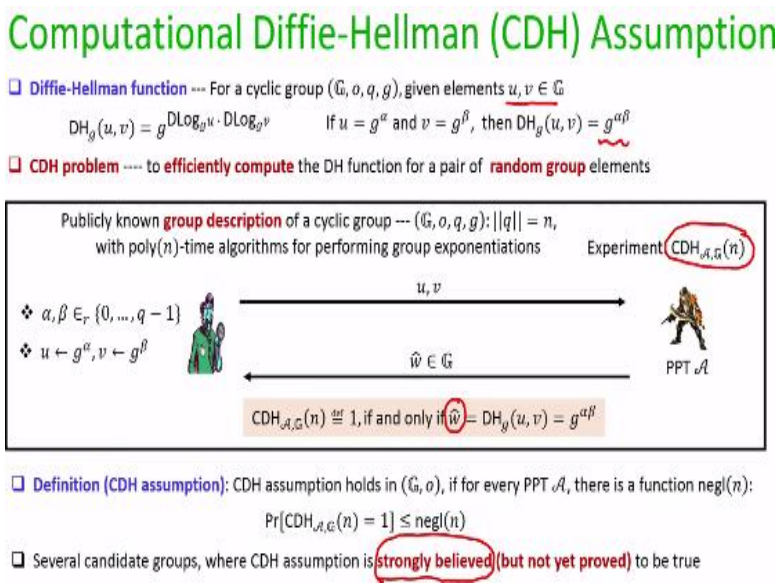
Namely it means that each element in the group G requires n number of bits for the representation and we also assume that we have polytime algorithms for performing group operations and exponentiation and poly time where the polynomial in the number of bits that you read to represent the value q. So the steps or the rules of the experiment as is follows. So the challenger or the experiment picks a random index α in the range 0 to q – 1.

And once it picks the index α it computes the value u which is g^{α} and gives it to the challenger. So before I proceed here I would like to stress that this whole experiment this discrete log experiment I am formulating assuming a multiplicative group representation, but

you can imagine that the corresponding experiment is also there for a group cyclic group where the underlying operation is additive.

So since the index α is randomly chosen from the set 0 to q – 1 that automatically means that the element u which is thrown as a challenge to the adversary is also a random group element okay and now the challenge for the adversary is to find out the discrete log of u namely the value α. So it has to output a value α' and in the range 0 to q –1 and the rules of the experiment says that we say that the adversary has won the experiment or the output of the experiment is 1 if and only if the α' which adversary has submitted indeed is a discrete log of the value u, namely if $g^{\alpha'} = u$ holds and we say that the discrete log assumption holds in the group (G, o) if for every polytime adversary participating in this experiment there is some negligible function such that the probability adversary wins the experiment or the output of the experiment is upperbounded by that negligible function and there it turns out that there are several candidate groups, where indeed the discrete log assumption is strongly believed to be true. I stress that it is only believed that in those candidate groups the discrete log problem is indeed difficult to solve, but it is not yet formally proved that means over the last 30, 40 years even after performing rigorous research no one is able to come up with polytime algorithm or efficient algorithm which can solve or compute a discrete log for randomly chosen elements from those candidate groups. That is why we strongly believe that the discrete log problem is indeed hard in those groups and that is why we call this whole problem as the discrete log assumption.

(Refer Slide Time: 04:42)



Now let us see a related *assumption* which we call as the computational Diffie–Hellman assumption or CDH assumption and for that let us first introduce the notion of Diffie–Hellman function. So imagine we are given the description of a cyclic group where the generator g is known and the size of the group is q and say we are given elements u and v from the group. I beg your pardon, the Diffie–Hellman function of the input u, v with respect to the generator g $\mathrm{DH}_g(u, v)$ is defined to be the value g to the power discrete log of u multiplied by discrete log of v : ($g^{\mathrm{DLog}_g u.\mathrm{DLog}_g v}$).

Again this Diffie–Hellman function is defined assuming that the underlying group operation is multiplicative, but you can imagine the corresponding Diffie–Hellman function where the underlying operation in the group is an additive operation. If you look closely then the way we have defined this Diffie–Hellman function then it turns out that since u and v are elements of the group then u can be expressed as some g^{α}.

And in the same way the element v can be expressed as some g^{β} that means if I say the way the Diffie–Hellman function on the input u, v it is nothing, but g to the power the discrete log of ($g^{\alpha\beta}$). Now what exactly is the computational Diffie–Hellman problem or CDH problem well the problem here is that you are given the description of a cyclic group and the generator and you are given a pair of randomly chosen elements from the group and the challenge or the problem that we are interested to solve is to compute the Diffie–Hellman functions with respect to those pair of inputs.

And the CDH assumption basically states that there exist candidate groups or there exist a group where the CDH problem is indeed difficult to solve in polynomial amount of time. So we formally model this requirement by an experiment which we call as the CDH experiment played between a computationally bounded adversary and an experiment or a verifier what is publically known is the description of the group, the group operation, the generator and the size of the group.

And we assume that the number of bits that we need to represent the value q is n, where n is security parameter. The rules of the experiment are as follows. The challenge prepares the challenge for the adversary by picking random indices α, β in the range 0 to q – 1 and computes the element g^{α} and g^{β}. So you might be wondering that is it efficient to compute.

Is it known how to efficiency compute g^{α}, g^{β} if given α and β and g yes it turns out the answer is yes that is what I am assuming here that for the underlying group we have polytime algorithms for performing the group exponentiation due to lack of time I am not discussing those exact algorithms how to compute g^{α} in poly of n amount of time.

And you can see the book by Katz & Lindell or any standard reference where you can see the details of how exactly to compute or perform group exponentiation in polynomial amount of time. So once the experiment or the challenger has prepared or computed the elements u, v the (u, v) pair is thrown as a challenge for the adversary and the challenge for the adversary is to compute the Diffie–Hellman function on this input pair (u, v).

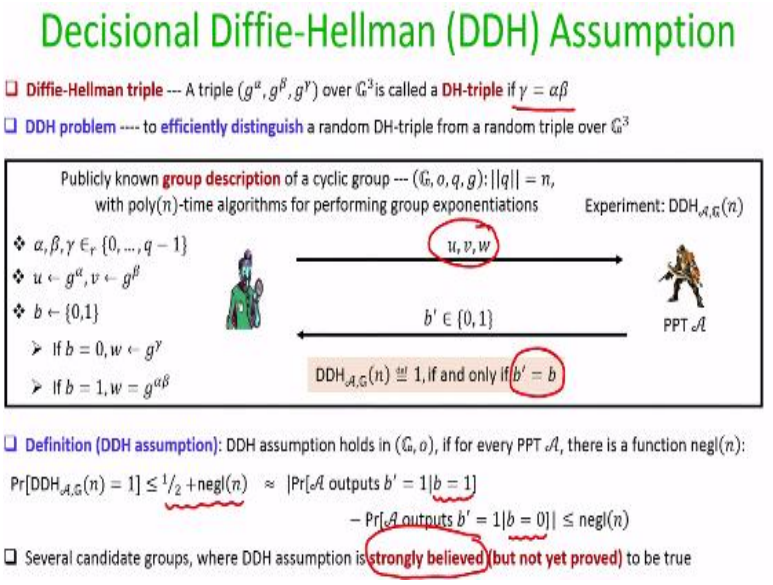
So since that output of the Diffie–Hellman function is also going to be a group element basically the adversary has to submit an element or output an element from the group which I denote as this $\hat{w}$ and the rules of the experiment says that we say that the adversary has won the experiment or the output of the CDH experiment is 1 if and only if the output $\hat{w}$ submitted by the adversary is exactly the same as the value of the Diffie–Hellman function on the input pair (u, v) namely it is same as $g^{\alpha\beta}$.

So that means the challenge here for the adversary is to compute $g^{\alpha\beta}$ without actually knowing α and β which are randomly chosen by the challenger here and the CDH assumption basically says that we say that the CDH assumption holds in the group (G, o) if for every polytime adversary participating in this experiment with respect to the group (G, o) the probability that the adversary can win this experiment or solve the CDH problem is upper bounded by some negligible function in the security parameter.

And it turns out that there are several candidate groups where the CDH assumption is strongly believed to be true again I stress that it only strongly believed that there exist no polytime adversary who can significantly solve, who can solve the CDH problem with significant probability, but it is not yet formally proved it is just an assumption. Over a period of time even after rigorous attempts no polytime algorithm have been obtained.

And that is why we believed that in those candidate groups the CDH problem is indeed difficult to solve and that is why we say that in those candidate groups the CDH assumption holds.

(Refer Slide Time: 09:54)



And finally let us see another related problem or assumption which we call as the DDH or Decisional Diffie–Hellman assumption and for that let me introduce the definition of Diffie–Hellman triple. So a triplet of element from the group which I say denote it as g^{α}, g^{β} and g^{γ} is called as a Diffie–Hellman triple if γ = the product of α and β.

Again this definition of Diffie–Hellman triplet is with respect to a cyclic group where the underlying operation in multiplicative, but this definition can be modified with respect to a cyclic group where the underlying operation is the addition operation. Now the DDH problem is to efficiently distinguish a randomly chosen Diffie–Hellman triplet from a randomly chosen triplet over the group.

That means an adversary or an algorithm will be given a triplet which could be either a Diffie–Hellman triplet or it might be an arbitrary randomly chosen triplet and it has to distinguish whether it is seeing a Diffie–Hellman triplet or a randomly chosen triplet and this requirement is formalized by an experiment which we call as the DDH experiment and the experiment is played between a computationally bounded adversary and arbitrary challenger or the experiment where the public information which is available is the description of a cyclic group namely the operation o the size of the group and the generator and the challenge for the adversary is prepared as follows. The challenger pick some random indices α, β, γ

uniformly randomly in the set 0 to q -1 and compute the first 2 components of the triplet which is going to be thrown as a challenge to the adversary.

Namely u is computed as g^{α} and v is computed as g^{β}. Since α and β are randomly chosen it automatically implies that the elements u and v are also random elements from the group and now to generate the third component of the challenge which is going to be thrown to the adversary the challenger picks or throws a uniformly random coin with probability half it could be 0 with probably half it could be 1.

If the coin toss is 0 then the third component of the triplet namely w is said to be g^{γ} that means w is a uniformly randomly element independent of u and v whereas if the coin toss is 1 then w is set to be $g^{\alpha\beta}$ and now once the w is decided the triplet u, v, w is thrown as a challenge to the adversary and the challenge for the adversary is to decide or distinguish or identify whether it is seeing a Diffie–Hellman triplet.

Or whether it is seeing a truly random triplet over the group; that means it has to identify whether the triplet u, v, w is generated according to the method b = 0 or whether it is generated according to the method b = 1 so adversary submits its response namely it output a bit b’ and the rules of the experiment and the definition of the experiment is that we say that adversary has won the experiment or the output of the experiment is 1, if and only if the adversary is correctable to identify whether it has seen a triplet by the method b = 0 or by the method b = 1 namely it has correctly identified b’ = b and we say that the DDH assumption holds in the group (G, o) if for every polytime adversary participating in this experiment there exist some negligible function such that the probability that the DDH experiment output is 1 with respect to that adversary is upper bounded by half + negligible.

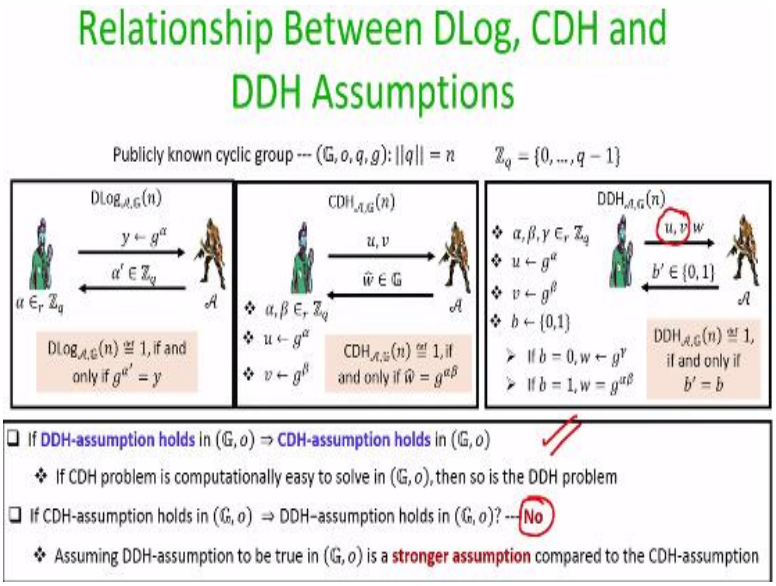
So notice that for the DLog experiment and for the CDH experiment the definition was that the output of the experiment is 1 should be upper bounded by some negligible function because there the goal of the adversary was not to distinguish something versus something. If the challenge for the adversary was to compute something, but in this DDH experiment or in this DDH problem the goal of the adversary is to distinguish one kind of triplet against another kind of triplet.

And that is why we are giving an indistinguishability type definition and that is why the definition is that we say the DDH assumption holds only if no polytime adversary can distinguish apart a Diffie–Hellman triplet from a non Diffie–Hellman triplet except with probably half + negligible. Equivalently the same condition can be put in this way. We can say that the DDH assumption holds in the group if it does not matter whether the triplet which is thrown as a challenge to that adversary is generated by method b = 1 or by the method b = 0.

The response of the adversary is almost identical namely say b' = 1 in both the cases except with some negligible probability and it can be proved that both this conditions are equivalent to each other. Again it turns out that there are several candidate groups where the DDH assumption is strongly believed to be true and again I stress that it is only strongly believed that in those groups the DDH assumption or the DDH problem is difficult to solve it is not mathematically proved.

It is only because we do not have any polytime algorithm till now for solving the DDH problem in those groups that is why we believe that the DDH problem is indeed difficult to solve in those candidate groups.

(Refer Slide Time: 15:26)



So now let us see the relationship between the DLog assumption, CDH assumption and DDH assumption. So for your reference I have written the experiment for the DLog assumption with CDH assumption and DDH assumption and in all this experiments the public

information is the description of a cyclic group again for simplicity I assume that the underlying operation is multiplicative operation.

The group size is q and the generator is g which is publicly known and a number of bits which is used to represent the value q is n and I use the notation Z_q to denote the set 0 to q – 1 right. So in all this experiment whenever the challenger was picking or trying to generate a uniformly random element from the group it was basically picking an random index from the set 0 to q – 1.

All those steps where the challenger was picking a random index from the set 0 to q – 1 can be represented as if the challenger is picking an index randomly from the set Zq. So now let us first see the relationship between the DLog assumption and the CDH assumption. It turns out that if the CDH assumption holds in the group G then it implies that the DDH assumption also holds in that group.

Put in other way if you see the contrapositive of this implication if the DLog problem is indeed easy to solve computationally easy to solve in the group then using that algorithm it is very easy for any polytime adversary to solve the CDH problem as well in that group because if you can computationally solve if you can compute the DLog of a random element in poly amount of time and then if you use that algorithm in the CDH experiment, then using that algorithm, the adversary, A can compute the DLog of u and it can compute the DLog of v namely it can extract out the values α and β in poly amount of time and once it knows α and β then itself can compute $g^{\alpha\beta}$ correctly. That means $\hat{w}$ will be indeed the output of the Diffie–Hellman function on the random pair of inputs u, v.

That means if DLog problem is easy then so is the CDH problem. So that proves the implication in the first direction. Now what about the implication in the other reverse direction that means can we say that if the DLog assumption holds in the group then it also implies that the CDH assumption holds in the group as well and interestingly the answer is that we do not know anything about this fact.

In fact, we strongly believe that there might be groups where the CDH problem is easier to solve even though the DLog problem is difficult to solve. The reason for this is that if you see the description of the CDH experiment of the CDH problem the goal of the adversary is to

compute $g^{\alpha\beta}$ where it does not know α and β. One way of computing $g^{\alpha\beta}$ could be to compute α from u and β from v namely solving the discrete log problem, but that need not be the only way by which an adversary or a polytime algorithm could attempt to compute the value of $g^{\alpha\beta}$. There might be some shortcut or some other way to compute $g^{\alpha\beta}$ without actually computing α and β and that is why it might be the case that even though in your group the CDH problem is easier to solve the discrete log problem might be still difficult to solve.

And that is why assumption wise we say that CDH making an assumption that a CDH problem is hard to solve in your group is a stronger assumption compared to making the assumption that the DLog problem is difficult to solve in your group because it looks like that CDH problem is relatively easier to solve compared to the discrete log problem right. So assumption wise the CDH assumption is a stronger assumption.

Because the difficultly wise the CDH problem might be easier to solve compared to the discrete log problem. So that is a relationship between the discrete log assumption and the CDH assumption. Now let us see the relationship between the CDH assumption and the DDH assumption. So it turns out that if in your group the DDH assumption holds then the CDH assumption also holds and this can be proved easily by a contrapositive.

Specifically, if you have a polytime algorithm which can solve the CDH problem with significant probability then using that algorithm it is very easy to even solve any instance of DDH problem in that group. Basically what the DDH solver has to do is he has to, given the pair u, v, w what it has to do is it has to invoke the CDH solver on the pair u, v and get the value of the output of the Diffie–Hellman function on the input pair u, v.

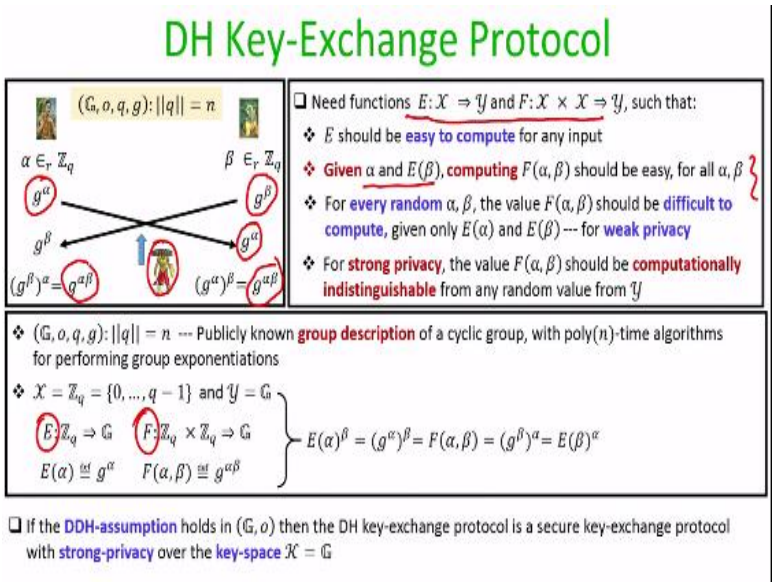
And compare that output with w and accordingly the DDH solver can decide whether it is seeing a Diffie–Hellman triplet or a randomly chosen triplet over the group so that is the implication in one direction now what about the implication in the other direction. Can we say that if the CDH assumption holds in the group then so is the DDH assumption and the answer in this case is clear no.

It turns out that we have some specific groups where we know how to solve the DDH problem in poly amount of time, but even though we know how to solve a DDH problem in poly amount of time in those groups we do not have yet any polytime algorithm to solve the

CDH problem with significant probability in those groups that means that making the assumption that the DDH problem is difficult to solve in a group is a stronger assumption compared to the assumption that a CDH problem is difficult to solve in that group.

Because difficulty wise we feel that a DDH problem is easier to solve compare to solving the CDH problem that means if we consider these 3 assumptions discrete log assumption is the mildest assumption because difficulty wise solving the discrete log problem is the hardest problem whereas making the DDH assumption is the strongest assumption because difficult wise solving the DDH problem might be computationally less expensive compared to solving the CDH problem compared to solving the DLog problem.

(Refer Slide Time: 21:48)



So now we have the 3 assumptions the CDH the discrete log assumption, CDH assumption and the DDH assumption and now we are going to see the exact mathematical steps or the concrete steps of the Diffie–Hellman Key Exchange Protocol. So remember in the last lecture we have seen the steps of the Diffie–Hellman Key Exchange Protocol assuming that we are given some special function E and F right.

So just to summarize the requirement from the function E should be that it should be easy to compute, it should be one way and not only that given α and the value of the function E(β) even without knowing β it should be easy to compute F(α, β) if you know α and we had also seen that if you want to obtain a Diffie–Hellman Key Exchange Protocol with the notion of weak privacy then the requirement from the function E and F should be that if someone gives you the value of E(α) and E(β) then just by knowing E(α) and E(β) it should be difficult for

you to compute F(α, β) in its entirety, but if you want to achieve a stronger notion of secrecy namely strong privacy from the Diffie–Hellman Key Exchange Protocol then we need additional requirement from the function E and F namely we require that the value of F(α, β) should be computationally indistinguishable from any random value from the set y.

Even if you are given with the value of E(α) and with the value of E(β). Now let us see how exactly we can instantiate this functions E and F that we have in this abstract Diffie–Hellman Key Exchange Protocol. So the public setup that we assume that is available with the party is the description of a cyclic group and by the description I mean the group operation, the size of the group and one of the generators.

And we assume that number of bits that we need to represent the value q is n. So we instantiate the set x with the set Z_q namely the set 0 to q – 1 and the set y by the cyclic group G and we define the function E to be a function mapping the elements from the set Z_q to the group by the relationship that E(α) to be defined is defined as g^{α}.

Again for simplicity I am assuming that the underlying group operation in the cyclic group is a multiplicative group and that is why E(α) is defined to be g^{α} whereas if the underlying group operation would have been an additive group then E(α) would have α times g, right. So that is what is instantiation of my E function and I define my function F to be a 2 input function taking a pair of inputs from the set Z_q and giving me an output a group element where F(α, β) is defined to be $g^{\alpha\beta}$. And now you can see the way we have defined the function E and function F it satisfies one of the requirements namely if you are given the value of E(α) and you do not know α but you know β then just by raising E(α) to the power β : $(E(\alpha))^{\beta}$ you get the value of F(α, β).

And in the same way if you are given the value of just E(β) and you do not know β but you know α then by raising E(β) to the power α : $(E(\beta))^{\alpha}$ you end up getting the value of F(α, β) right. So now let us see with how exactly using this instantiation of E and F the exact Diffie–Hellman Key Exchange Protocol will look like. So the public setup will be the description of the cyclic group.

And sender will independently pick an index α from the set Z_q and in the same way the receiver is going to pick a random index β from the set Z_q. Sender is going to compute a function output $E(\alpha)$ namely it will prepare or it will compute a value g^{α} and in the same way the receiver will compute the value g^{β}. So if we can imagine that g^{α} is sender's contribution to the final key which is going to be established between the sender and the receiver.

And in the same way the component g^{β} you can imagine as if it is the receiver's contribution towards the final key which is going to be established between the sender and the receiver. Now both sender and receiver are going to exchange over a public insecure channel their respective contributions towards the final key and once sender receives reciever's contribution, namely g^{β} it has to raise it to its own contribution namely α to obtain the final key which is going to be $g^{\alpha\beta}$ and in the same way receiver once it receives sender's contributions g^{α} it raises it to the own contribution β to obtain final key $g^{\alpha\beta}$ okay. So it turns out that if we want to have weak privacy from this Diffie–Hellman Key Exchange Protocol then we need a CDH assumption to be true in the underlying cyclic group.

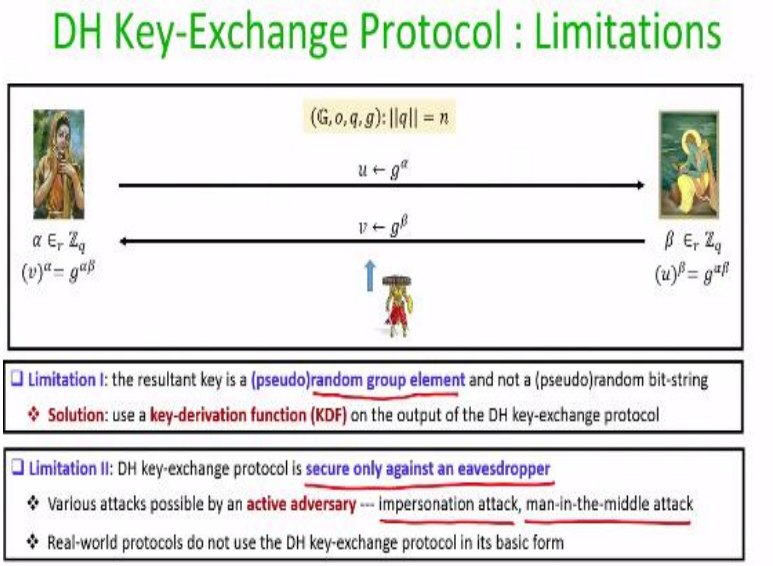
That means if it is ensured that in the underlying cyclic group which the sender and receiver are using to instantiate this Diffie–Hellman Key Exchange protocol holds if the CDH problem is difficult to solve in that group then indeed an eavesdropper who learns g^{α} and g^{β} where α and β are randomly chosen, cannot figure out what exactly is $g^{\alpha\beta}$.

Because that precisely is what an instance of the CDH problem right. On the other hand, if you want that from the same Diffie–Hellman Key Exchange Protocol we achieve the notion of strong privacy then we require that in the underlying group the DDH assumption should be true, namely an eavesdropper who has seen g^{α}, g^{β} from its viewpoint the final output of the sender and receiver, namely the key $g^{\alpha\beta}$ should be computationally indistinguishable from any random element from the underlying group g and that is precisely is a DDH assumption because if the eavesdropper based on the g^{α} and g^{β} is able to learn anything about $g^{\alpha\beta}$ namely it can now distinguish apart this is the resultant $g^{\alpha\beta}$ from any other random element from the underlying group, then that means if an eavesdropper knows in polynomial time how to significantly distinguish a Diffie–Hellman triple from a non Diffie–Hellman triple, but that goes against the assumption that the DDH assumption is DDH problem is difficult to solve in the underlying group. So you can see that just by using different assumptions we get different

notions of privacy from the concrete Diffie–Hellman Key Exchange Protocol which we are now instantiating using this function E and this function F.

For weak privacy we need the CDH assumption to be true whereas for the strong privacy we need the DDH assumption to be true.

(Refer Slide Time: 28:55)



So now we have seen the exact steps of the Diffie–Hellman Key Exchange Protocol. So even though it allows the sender and a receiver to communicate over a publicly known insecure channel with no pre shared information and agree upon a common random key it turns out that there are certain limitations of this key exchange protocol. The first limitation is that the resultant key which sender and receiver are going to output here is not a random or a pseudo random bit string.

But rather it is a pseudo random group element whereas we would like sender and receiver to agree upon a pseudo random bit string because once which they can use for any symmetric cryptographic primitive right. So remember in any symmetric cryptographic primitive we make the assumption that both sender and receiver start with a pre shared common random key which is a bit string which is known only to the sender and the receiver.

But by running this Diffie–Hellman Key Exchange Protocol a sender and the receiver can only agree upon a pseudo random group element not on a pseudo random bit string. So a potential solution to solve or get around this problem is to use a key derivation function

which we had seen during our discussion on the hash function and we can use any of the standard key derivation function based on the hash function.

And assuming that the resultant pseudo random group element which sender and receiver are going to output at the end of the Diffie–Hellman Key Exchange Protocol has a sufficiently large entropy then by applying the key derivation function on that pseudo random common output both sender and receiver can locally obtain a pseudo random bit string which they can now use as a key for any symmetric key cryptographic primitive.

And we will encounter this idea again and again later on when we will discuss the notion of publically crypto system and hybrid crypto system. The second limitation which is there in the Diffie–Hellman Key Exchange Protocol is that it gives you security only against an eavesdropper who monitors the communication between the sender and the receiver. It turns out that if our adversary is an active adversary where it can intercept packet or where it can change the contents of the messages communicated between the sender and the receiver then it can launch various kind of attacks like impersonation attack, man in the middle attack and so on. That is why in the real world protocol we do not use the Diffie–Hellman Key Exchange Protocol as it is in its basic form. We make modifications on top of the Diffie–Hellman Key Exchange Protocol to ensure that we take care of even an active adversary.

So that brings me to the end of this lecture. Just to summarize in this lecture we have introduced various cryptographic assumptions in cyclic group namely the discrete log assumption, the CDH assumption and the DDH assumption and based on these assumptions we have seen the concrete steps of the Diffie–Hellman Key Exchange Protocol. Thank you.