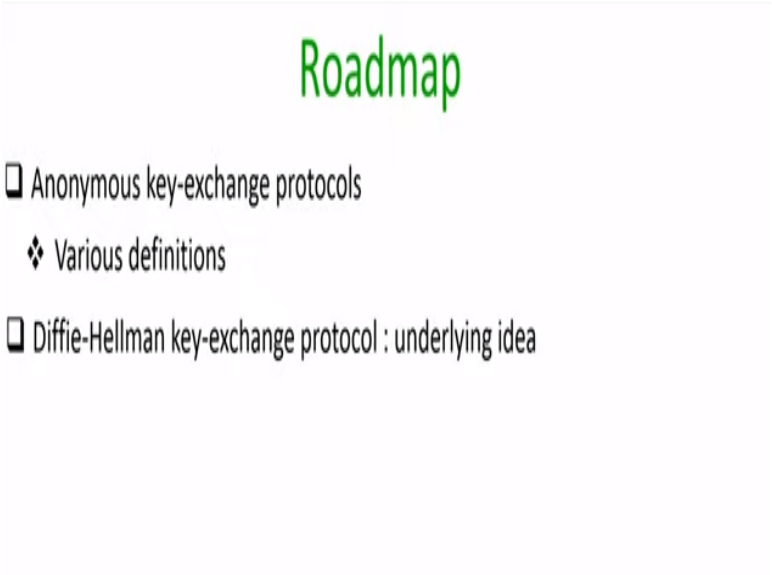
Foundations of Cryptography
Dr. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bangalore

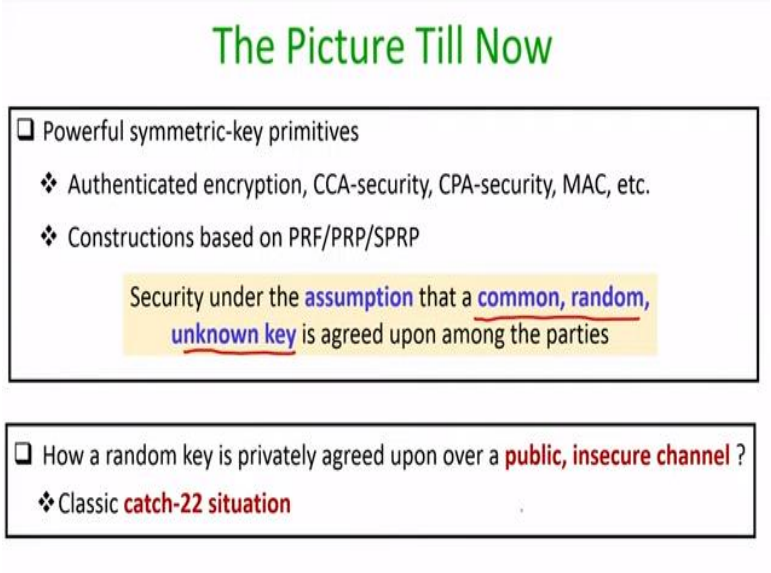
Lecture – 36
Key Exchange Protocols Part 1

(Refer Slide Time: 00:35)



Hello everyone, welcome to this lecture the plan for this lecture is as follows, in this lecture, we will introduce the notion of anonymous key exchange protocols. We will see the various definitions and we will see the underlying ideas involved behind the Seminole a key exchange protocol due to the Diffie and Hellmann.

(Refer Slide Time: 00:47)



So just to recap the picture till now is as follows till now we had seen several powerful symmetry key primitives both in the passive world as well as in the active world. We have seen various notions of security like authenticated encryption CCA security, CPA security we had seen other primitives for integrity and authentications like message authentication code and we had seen constructions of all this notions of security enabled.

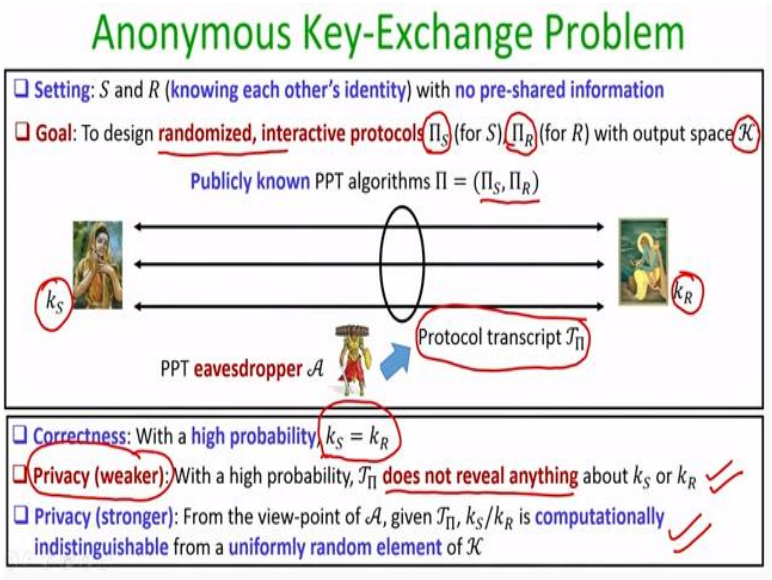
We had seen constructions giving you this this kind of security whether constructions are based on pseudo random function pseudo random permutation strong pseudo random permutations and so on. So a basic fact about all the constructions that we had seen until now is that the security for all this cryptographic primitives that we have seen holds and that assumption that a common random unknown key is already agreed upon between the parties specifically the sender and the receiver.

So now the question that we want to address here is that how a common random keys privately agreed upon over a public insecure channel right? So till now we were assuming that somehow the key agreement has already happened and given that the key agreement has already happened we saw how to go how to design authenticated encryption schemes CCA secure encryption scheme CPA secure encryption schemes and so on.

But now we would like to address the main problem. We would like to address the question that how at the first place itself a random key is privately agreed upon over a public and insecure channel and this sounds like a classic catch 22 situation right? So because if you have a mechanism where sender and receiver can securely exchange a unknown key which is known only to the sender and the receiver over a public insecure channel then using that mechanism, they at the first place itself can do the secure communication also right.

So what exactly catch 22 situation means where you have a scenario where say for example you a fresh graduate comes out of the college and applies for a job and when the candidate faces the job interview, he faces the question that do you have any experience or not? But at the first place the candidate will get experience only if the job is given to the candidate. So that is what we mean by catch 22 situation. And we are exactly facing the same scenario here as well we know how to do secure communication assuming that we know how to securely exchange the key over an insecure channel.

(Refer Slide Time: 03:16)



So that brings us to the problem of anonymous key exchange. So let us define what exactly is the anonymous key exchange problem. So the setting is as follows we have a sender and the receiver, and we assume that somehow, they know each other's entity but did not have any pre shared information. So you might be wondering that how it is possible for a sender and the receiver who are meeting for the first time that they know each other's entity.

Later on we will remove this assumption as well but for the moment to make the description of the anonymous key exchange problem simpler let us assume that both sender and resume the receiver know each other's entity. So the goal here is to design a pair of protocols which I denote as pi S pi sub S for the sender and pi sub R for the receiver which are now interactive protocols.

And the output space of this protocol will be some fancy K and the goal is to design such a pair of protocols one for the sender and one for the receiver such that according to the individual protocols sender and receiver communicate over an insecure channel and at the end of the protocol at the end of their respective protocols sender and receiver output some respective keys so the output of the sender I denote as K sub S and output of the receiver I denote as K sub R.

And we assume in this problem definition that we have an eavesdropper a computationally bounded eavesdropper who is getting access to the entire communication that is happening between the sender and receiver. So the adversary is aware of the pair of protocols using which the sender and a receiver are interacting namely it knows the steps of pi sub S and pi sub R and it also gets access to the information that is exchanged between the sender and the receiver which we call us to protocol transcript.

Which I denote as tau sub pi and notice that this protocol transcript is going to be a randomized variable because the information which sender and receiver are going to exchange, they will depend upon the internal randomness with which the sender and receiver are going to invoke the respective protocol pi sub S and pi sub R. So it is not the case that sender and receiver will exchange the same set of values every time they invoke this protocol pi because they are invoking a randomized protocol.

Now the properties that I require from this pair of protocols pi sub S and pi sub R are as follows. The first property is the correctness probability we say start with very high probability the individual outputs of the sender and the receiver should be the same namely output case sub S and output case sub R should be same. And now we can have 2 variants of security which we can demand from this protocols by pi sub S and pi sub R.

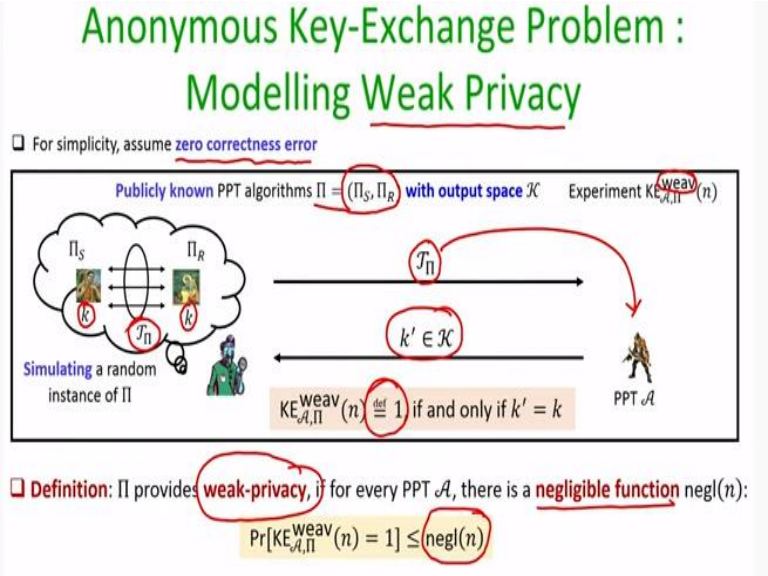
The first definition is a weaker form of privacy which I call as weak privacy which requires that with very high probability a computationally bounded adversary even after knowing the protocol transcript does not learn anything about the output of the sender and the receiver. So in here I am not demanding security in the indistinguishability sense here the demand is in the sense that either the adversary should not learn the entire kS and the entire kR it is fine if the adversary learns something about kS and sub R, kS and k sub R. But the requirement here is that as an entity the outputs of the sender and the receiver should not be learn to the attack.

Whereas we can go for a higher notion of security which I call that stronger privacy which demands that from the viewpoint of that adversary even if the adversary has access to the protocol transcript from the viewpoint of the adversary the respective outputs of the sender and the respective output of the receiver should be computationally distinguishable from a uniformly random element of the output keys fancy K.

So that is a more stronger notion of secrecy because here because if you go for the stronger notion of secrecy then it is not allowed at adversary learn something about the respective bits of case sub S and case sub R from the viewpoint of that adversary the respective outputs of the sender and the receiver could be any candidate element from the underlying key space fancy K right

So we will see construction satisfying both the weaker form of privacy and the construction satisfying the stronger form of privacy you might be wondering that why exactly we care to achieve weaker form of privacy. So that will be clear later on looking ahead we will see constructions achieving we call for more privacy based on cryptograph consumptions which are mild. Whereas if you want to achieve key exchange protocols which achieve stronger notion of privacy then we have to go for cryptographic assumptions which are slightly stronger.

(Refer Slide Time: 08:05)



So we had seen intuitively what exactly privacy weaker privacy and stronger privacy means. So now let us go ahead and model this requirements formally by an indistinguishability based game. And while giving the indistinguishability definition for simplicity we assume that we are considering key exchange protocols which has 0 correctness. That means with this with probability 1 output of the sender and output of the receiver will be the same, So we are considering that kind of key exchange protocols.

So let us see first how to model the weak privacy and remember the requirement of the week privacy start even if that was her he sees the whole transcript exchange between the sender and the receiver the resultant output key which are out which are which is obtained by the sender and the receiver should not be learned or should not be known to that adversary in its entirety and that remodeled by this game.

So this game we call as key exchange and a superscript weav eav denotes weak eavesdropping or weak privacy here the game is played between a challenger and a poly time adversary. And what the challenger or the experiment does is basically it simulates a random instance of the key exchange protocol pi. So the protocol pi is basically the pair of protocols by service and by pi sub S and pi sub R where the pi sub S protocol is going to be invoked by the sender and the pi sub R protocol is going to be received by the receiver.

But as a collection we call it as a protocol pi. So the challenger basically simulates a random instance of the protocol pi by playing the role of the sender and a receiver and its mind with their respective coins as per the protocol pi and it denotes a transcript tau. Now what it does it throws a challenge to the adversary namely the transcript and this modern sub fact that in the real world the adversary set an adversary sitting between the sender and the receiver would have observed a transcript which are generated which is a generated by the sender and receiver by running the protocol pi

Now the challenge for the adversary is that by seeing this transcript tau it has to figure out what exactly is the value of key k which sender and receiver would have obtained by running by the respect to this protocol tau. So basically that adversary has to respond with a key from the key space which I denote as k dash and the security definition and the way we defined output of the experiment is as follows.

We say that the experiment we say that adversary has won the experiment if I know which also which was also denoted by saying that output of the experiment is 1 if and only if the guess of the adversary k dash is exactly equal to k that means output of the adversary the adversary A without knowing the internal randomness of the sender and the internal randomness of the receiver and by just observing the protocol transcript tau is able to come up with the exact key k which sender and receiver are going to obtain by running with respect to this protocol transcript tau.

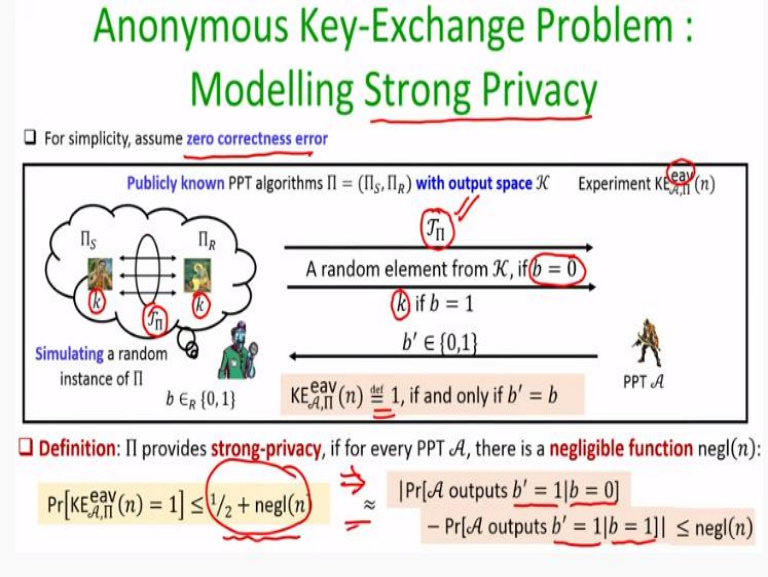
If that happens then we say that output of the experiment is 1 and the security definition is we say that the protocol pi the key action protocol pi has weak privacy if for any poly time adversary participating in this experiment, there exists some negligible function such that the probability that adversary means to experiment is upper bounded by some negligible function where the probability is taken over the random coins of the challenger.

Namely the random coins with which it is simulating the role of the sender in the receiver. So notice that here the adversary does not have to does not have to distinguish between something right? The goal of the adversary is to come up with the key which the sender and the receiver I

was going to obtain with respect to the protocol transcript tau. That is why the security definition will not have an expression of the form that adversary chances of winning the experiment is upper bounded by half plus some negligible function.

The goal of the adversary is to come up with the exact gave with which sender and receiver are going to obtain by running the protocol pi and which is consistent with the protocol transcript. Also in this definition we allow the adversary to win the game with some negligible probability because there is always a guessing adversary who can just guess some candidate k dash from the key space and with non-zero probability it may turn out the k dash is exactly equal to k.

(Refer Slide Time: 12:31)



So now let us see how we can model strong privacy and remember the goal of strong privacy is that an adversary who monitors the transcript exchange between the sender and the receiver should not be able to distinguish the resultant key which sender and receiver are going to obtain from any uniformly random element from the key space. And again for defining this modeling the strong privacy we assume key exchange protocols where there is 0 correctness error.

This is again without loss of generality its very straight forward to incorporate this condition as well in the security definition. So now the experiment is called eav because now we are not modeling weak privacy. We are now modeling strong privacy here and the rules of the game is as follows rules of the game are as follows the adversary is waiting for a challenge here and the

challenge is again generated more or less in the same way as it was generated in the experiment for weak privacy.

Namely the challenger here plays the role of the sender and receiver in its mind and invokes an instance of the protocol pi sub S and then for the sender and invoke in the instance of the protocol pi sub R for the receiver with their respective randomness and it generates a protocol transcript tau. That protocol transcript is now given as a challenge to the adversary. So this model stuff fact that that was a research thing between the sender and a receiver would have eavesdropped and obtained the protocol transcript tau pi.

And now since we are trying to modeling the indistinguishability base to motion of security for the queue in the context of key action protocol the challenger additionally throws a challenge as follows. It tosses a fair coin with probability 1/2 it could be 0 or with probability 1/2 it could be 1 and now apart from the transcript right depending upon whether the coin toss is 0 or whether the coin toss is 1 the challenger either submits a random element from the key space to the adversary or the key k which the sender and receiver would have obtained by running the protocol instance in the challenger's mind.

Now adversary is does not know whether the value from the key space that it is seeing is a random element from the key space or whether it is a key which the sender and the receiver would have obtained by running the protocol and by running the protocol pi and obtaining the transcript. So the challenge for the adversaries to generate whether he is in the method b = 0 or whether it is in the method b =1 and it submits this response namely a bit.

And the definition here is we say that the adversary wins the experiment which also denoted as the output of the experiment is 1 if and only if adversary A could correctly identify whether he is seeing an element as per the method b = 0 or whether he is seeing an element from the key space as for the method b = 1 right and the definition of strong privacy here is we say that a key exchange protocol pri gives you a strong privacy for any poly time adversary participating in this experiment there is some negligible function such that the probability adversary wins the experiment is upper bounded by some half plus negligible function.

So now this is different from the this definition is different from the weak privacy because in the weak privacy the goal of that adversary was to come up with the exact k which is consistent, or which would have been obtained by the sender and receiver as per the transcript tau. But now here the goal of that adversary is to distinguish the k which sender and receiver are going to obtain as per the protocol transcript tau from a uniformly random element from the key space.

So that is fine now we are going now we are having a condition which similar to the indistinguishability based definition and again we are putting this condition half plus negligible because there is always a guessing strategy by the adversary with which you will and with and the guessing strategy of the adversary will be just to guess whether he is in the method b = 0 or whether it is in the method b =1.

And the success probability of that guessing strategy is 1/2. Apart from that we are willing to let the adversary with this experiment with some negligible function because we are in the computational world and equivalent formulation of this notion of strong privacy is that we say that the protocol pi gives you a strong privacy if for any poly time adversary participating in this experiment.

The distinguishing advantage of the adversary is upper bounded by some negligible function in the security parameter. That means it does not matter whether the challenger has generated the challenge by the method b = 0 or whether he has generated the challenge by method be equal to one. In both the cases the response of the adversary should be almost as same cb dash = 1except with some negligible probability and we can prove that both does conditions are equivalent to each other.

Namely if you have a key exchange protocol satisfying the first condition then it also implies that the template satisfies the second condition and vice versa So depending upon our convenience we can use any of these two conditions.

(Refer Slide Time: 17:41)

So now that we have definitions of the prior we have those security definitions of key exchange protocols and now you might be wondering that whether in India its possible to design key exchange protocols where sender and receiver without having any without having any pre shared information can do some communication or what a publicly known insecure channel and end up agreeing upon a key which is not known to any third party.

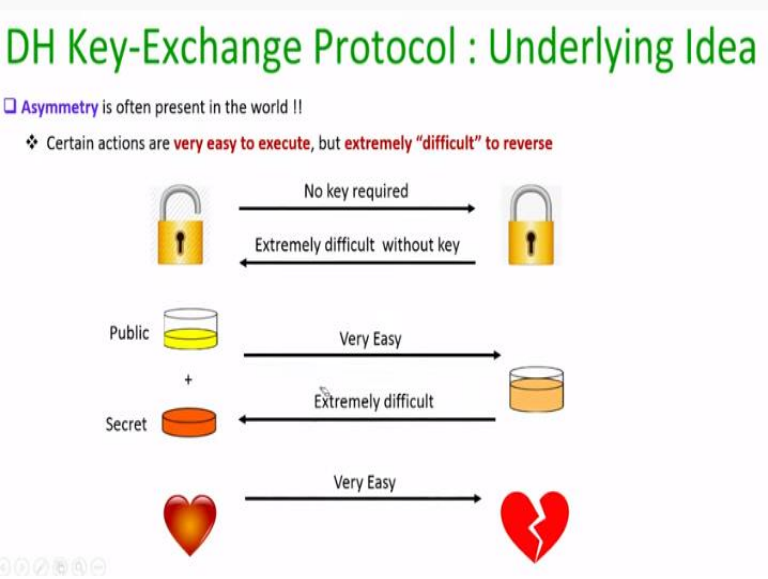
So on a very high level it might look like an impossible task because how it is possible that is unknown sender and receiver who just know their identity and have no pre shared information whatsoever can do public interaction and agree upon something which known only to them. But it turns out that we indeed we have some such key exchange protocols and its and the this is due to the base of this key exchange protocol is to pioneer work by the Diffie Hellman who are the first people to for the first pair of cryptographers who came up with their seminar key exchange protocol.

So in this lecture we are going to see the underlying idea based on which they are key exchange protocol was developed. So before that let me tell you some fascinating history about how exactly their key action protocol came into existence. So Whitefield e was a war very interested in solving the key exchange protocol or key action problem. And he strongly believed that indeed its possible for us sender and the receiver to do public communication and agree upon a common secret key.

And in 1974 he gave a seminar at IBMs Watson laboratory to a skeptical audience who were not buying his idea that indeed key exchange is possible over a public channel. The only positive outcome which came out of that seminar is that he came to know that its not only him there is another person a professor at Stanford called Martin Hellman who is also working on the same problem and trying to come up with public key action protocols.

And as soon as Diffie came to know about this he started some 5000 kilometer rod journey to meet and pair of with Martin Hellman and that is how they started their work on coming up with a protocol for solving the key exchange protocol and they work two years rigorously and finally they came up with their groundbreaking Diffie Hellman action protocol

(Refer Slide Time: 20:07)



So let us try to understand the underlying idea which is there in that if you had one key exchange protocol. So the basic idea behind their connection protocol is that there are several task in this world which have which have asymmetry or they are asymmetric in nature and asymmetry in the sense that they are very easy to execute. That means those actions certain actions are very easy to execute but extremely difficult to reverse.

So what I mean by that is imagine I am give I give you a padlock and an open state and if I ask you to lock it then you do not need any key. You just have to press the head of the padlock and

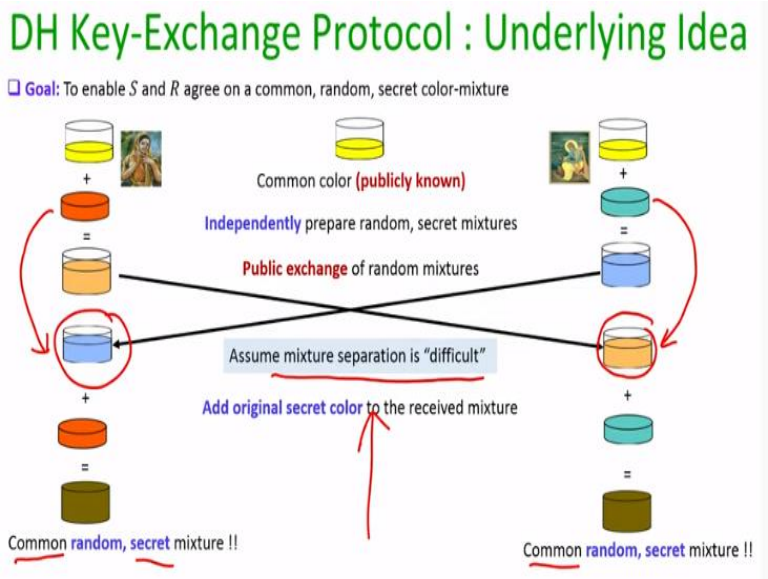
from the open state you can easily it to the locked state. But now if I give you the key padlock in the locked state and I ask you to take it back to the open state then it will be extremely difficult for you if you do not persist the key for the padlock.

So in that sense this you have an action here we are going from one state to another is extremely easy but going back from the obtain state back to the original state is extremely difficult. It is not impossible remember that I am saying its extremely difficult to go back or reverse the action if you do not know the key. There might be other methods to reverse that action without the key, but those alternatives will be extremely difficult.

In the same way consider this task that you are given a publicly known color and say if you want to prepare up secret mixture right then it is very easy for you to do that by taking that publicly known color and adding to adding to that existing color a secret color and then obtaining the mixture. So preparing the mixture is very easy for you. But if someone gives me this mixture and does not tell me what exactly was the secret color which was added to the public color to obtain this mixture then it will be very difficult for me to find out or separate out this mixture into its constitutes namely the publicly known color and a secret color which I have added here.

So in that sense repairing a mixture is easy but separating out the mixture to its individual component is an extremely difficult task. Again I stressed I am not saying it is an impossible status I am saying it is an extremely difficult task Its very time consuming and finally its very easy to break anyone's heart by just saying some bad words. But once someone's heart is broken then its very difficult to wins or convinced to win the back the love and the confidence of that person. In that sense this going from this state to this state is very easy but going back is extremely difficult right?

(Refer Slide Time: 22:50)



So based on this idea Diffie-Hellman thought of several ideas read how exactly they can solve the key action protocol and when they came about this idea that there are certain tasks which have asymmetry involved with them. Then based on that concept they thought of this idea of solving the key exchange protocol. So we are not going to see the exact mathematical details of the Diffie Hellman key exchange protocol which we that those details.

We will discuss in subsequent lecture I am just trying to give you the intuition that what exactly was done the Lang idea here So the goal here is for the sender and the receiver to agree upon a common random secret mixture color mixture which will be known only to the sender in the receiver who have no pre shared information to begin with. So here is how the protocol will proceed both sender and receiver will start with some common color publicly known.

And what they do is they individually prepare some random secret mixtures by adding a secret color independently picked and adding it to the publicly known color right? So both of them are independently taking the secret colors and preparing a mixture and once they obtain their respective mixtures what they do is they publicly exchange their mixtures with each other or some public channel right.

And when this mixture is our exchange, we assume here that the mixture separation is extremely time consuming extremely difficult it is a computationally heavy task that is an assumption I am

making here okay. Now once both sender and receiver obtains their respective mixtures what they do is do the mixtures that they have obtained from the other party. They add their respective secret colors with which they started the protocol.

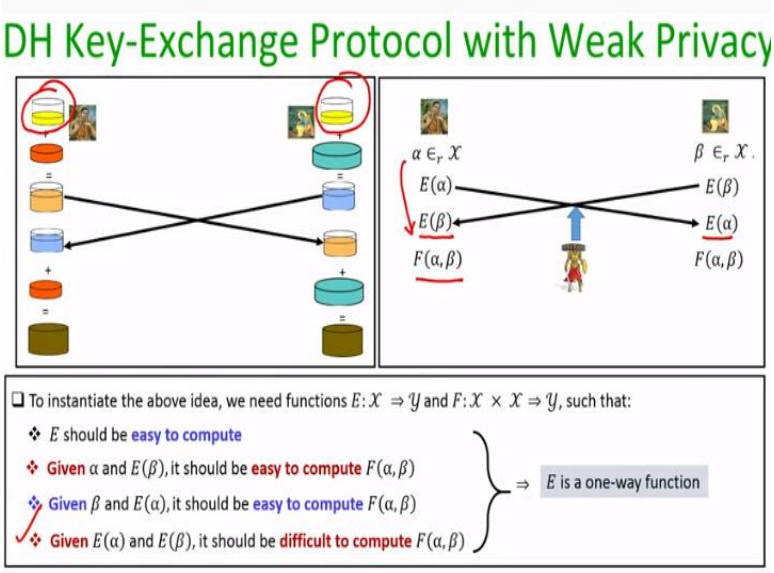
So for instance sender on receiving this mixture it takes a secret color and adds to it and in the same way the mixture that the receiver has obtained from the sender which is the senders mixture the receiver adds its own secret color and now you can see what both sender and receiver are going to obtain. Both sender and receiver are going to obtain a final common mixture because this final common mixture is a mixture of 3 colors the publicly known color, the senders secret component and a receiver secret component.

It does not matter in what order you mix them irrespective of the order in which you mixed them. The resulting mixture is going to be the same and that ensures that both sender and receiver are going to obtain a common mixture right and now why to secret? It is secret because we are assuming here that when sender and receiver are exchanging their respective mixture, mixture separation is extremely time consuming.

That means if there is an eavesdropper who is actually monitoring the communication happening between the sender and the receiver and it knows the entire description of the protocol step what the adversary does not know it would not know the exact secret components with the sender and the receiver have individually added. And even after seeing the mixture which sender and receiver are exchanging it will be difficult for the adversary to separate out or find out that respective things which the sender and receiver of addict.

And that ensures that the final mixture which sender and receiver are agreeing upon is already a secret color of a secret mixture which will not be known to any. So that is the fundamental idea behind that Diffie Hellman key exchange protocol. So of course this is an idea behind the key exchange protocol. Now we have to convert this idea into an algorithm.

(Refer Slide Time: 26:20)



Specifically, a mathematical algorithm. So what we are going to discuss now is how we can convert this idea of exchanging colors and coming up with a common secret mixture known only to the sender and the receiver into a key exchange protocol with weak privacy right? So this is I am retaining the blueprint of the key exchange protocol based on exchanging color mixtures and now what we have to do is we have to come up with an instantiation or replacement of each of the steps where colors are used mixtures are prepared and exchange and so on by some mathematical step.

So to instantiate the idea of an idea of what we need here is we need some special functions which I denote as big E and big and big F. So the function is from the state fancy x to fancy y, so it is a one input function whereas the function F is a two input function taking two inputs from the fancy set x and giving you an output from the fancy set y. The requirements from the functions are as follows.

So the function E should be easy to compute and the second requirement from the function E and function F is that if you are given any alpha from the domain of the function E and the value of the function E on any value beta then even without knowing the value beta just based on the knowledge of alpha and the function output of E on the value beta it should be easy to compute the value of the function on the input pair alpha, beta and this should hold for any alpha, beta.

That means if you know beta and if you know the value of the function E on an unknown input alpha then even without knowing alpha it should be easy for you to compute the value of the function or the alpha, beta. And finally since we are aiming for weak privacy, we require the following property we require that if someone knows the value of the function E on an unknown alpha and an unknown beta then it should be very difficult to find out the value of the function F on the value alpha, beta.

So these are the properties which I require from the function E and F and its easy to see that all together all these properties automatically implied at my function E should be one-way function. So remember we had already discussed what exactly is the one-way function long time back by one-way function informally is a function which is easy to compute on any value from their domain but computationally very difficult to invert for random values from the range of the function right?

So it is easy to see that all these implications implied at the function E should be a one-way function because if the function E is not a one-way function. Then given E of alpha it will be easy to compute alpha and given E of beta it will be easy to compute beta and automatically if you know alpha and beta and a description of the function F done you will be able to compute the value of F on alpha, beta which goes against the requirement that it should be difficult to compute F of alpha beta.

If you are just given the value of alpha and E of beta but not only, we require the function E to be one way. It turns out that we need some additional properties as well namely the property that we require here to start a few no alpha and the value of E on the input beta then even without knowing beta should be easy for you to compute F of alpha, beta and in the same way without knowing alpha given just E of alpha and beta should be easy for you to compute F of alpha, beta.

So these are now additional requirements on this one-way function later on we will see exact instantiations of this function E and F. But for the moment assume we have such functions E and F. Now let us see how exactly we can convert this color based protocol into a concrete key exchange protocol. So assume the public setup namely the public color here which is known

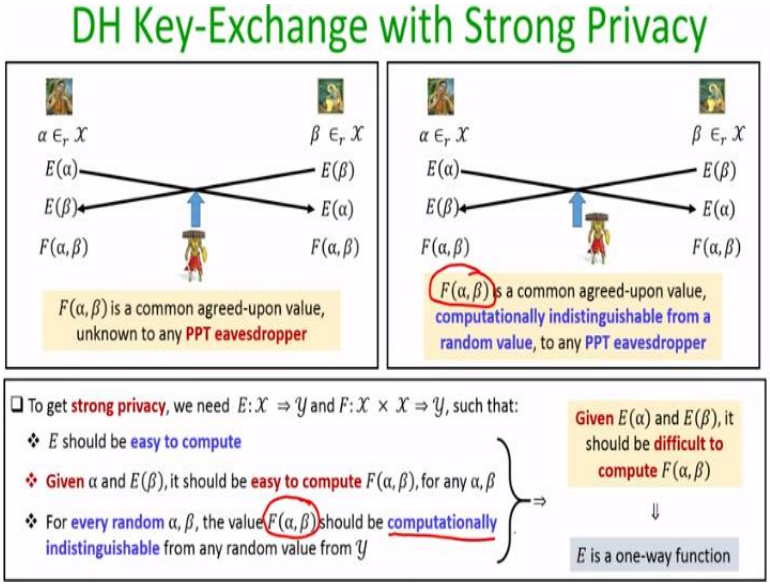
board to the sender and the receiver is nothing but a description of the function E and a description of the function F.

Now what the sender and receiver has to do they have to prepare their respective secret mixtures. So what they do is sender picks a random alpha from this x set and compute E of alpha and independently receiver picks a beta from the set x and compute E of beta and they exchange the values of your E of alpha and E of beta over a public channel. And now I am assuming that even if there is an adversary who sees the value of E of alpha and E of beta since a function is a one-way function it will be difficult for an adversary to find out what exactly I will find alpha and beta are that is what is the property Im assuming from the function E.

Now once E of alpha and E of beta are known to the sender and receiver respectively again by using the fourth property from the function E and F, we can see that its easy for both sender and receiver to compute F of alpha, beta. Why? Because sender is now knowing E of beta and it has alpha. So using the knowledge of E of beta and alpha it can obtain F of alpha, beta and in the same way the receiver its now learning E of alpha it is fine if he does not know alpha but given beta and E of alpha it can compute F of alpha.

And now you can see that F of alpha, beta is a common value from the set fancy Y which is known both to the sender and the receiver. But it will not be known in its entirety to any eavesdropper and that comes from the fact because the function is a one-way function.

(Refer Slide Time: 31:53)



Now if you want to obtain a key exchange protocol with a stronger privacy notion namely the strong privacy where the resultant key should be indistinguishable from any potential key from the key space right. Then we can retain the same blueprint of the defilement key exchange protocol that we have seen just now namely sender and receiver picking respective alpha. Beta and exchanging the values of E alpha and E beta.

What we have to just do now is to ensure that the resultant key namely or the resultant value and namely F of alpha, beta is a uniform is a computationally indistinguishable value from the viewpoint of an eavesdropper We need now some more stronger properties from the function E and from the function F namely the requirement of one wayness is still there The function E should be easy to compute but difficult to invert on random values from the domain and again given alpha and the value of E, E of beta it should be easy to compute F of alpha, beta for any alpha ,beta.

And now the stronger property that I require from the function E and F is that for every random alpha my beta the value of alpha, F of alpha, beta should be computationally indistinguishable from any random value y from any random value from the set fancy y. So this new requirement was not there when we were aiming to define the Diffie Hellman key exchange protocol with a weaker notion of privacy.

Because there the requirement was that adversary should not learn F of alpha, beta even if it knows E of alpha, E of beta. But now since we are aiming for a stronger privacy notion, we are putting additional requirements on the function E and F namely. We required that even if someone knows E of alpha E of beta the value of F of alpha, beta for that adversary or for that person should be computationally indistinguishable from any random value from the set fancy y right.

And if we assume that indeed our function E and F satisfies this requirement is additional requirement its easy to see that the same protocol which we had just seen giving us the weaker privacy notion ends up giving us a stronger privacy notion. We do not have to add additional step we just have to make stronger assumptions about the functions E and the functions F. So that brings me to the end of this lecture.

Just to quickly summarize in this lecture, we have introduced the problem of key exchange protocol we introduced the key exchange problem where the goal of the sender and the receiver is to interact publicly over an in secured channel with no pre shared information and end up with some common output which is known only to the sender and the receiver but unknown to any third party and we have introduced two notions of secrecy for the secrecy for such key exchange protocols namely the notion of weak privacy and the notion of strong privacy. Thank you.