**Lecture – 57**
Zero Knowledge Protocols Part I

(Refer Slide Time: 00:30)
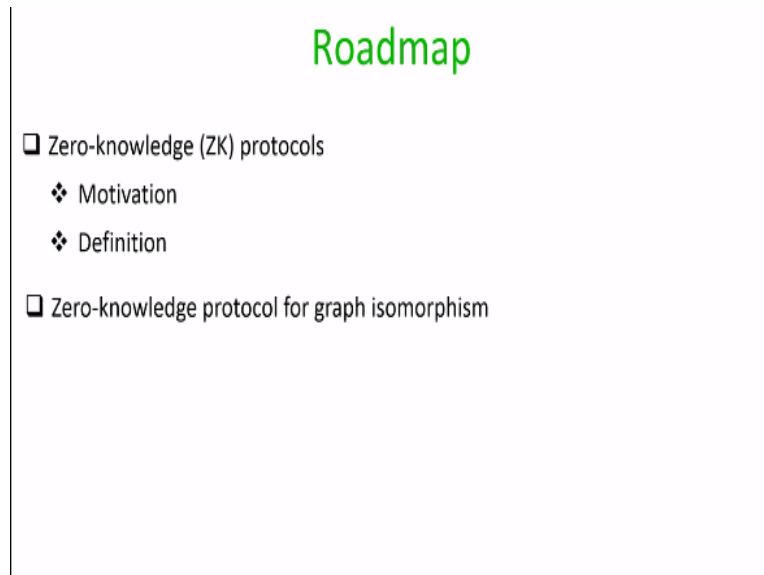
Welcome to this lecture. So in this lecture we will continue our discussion on interactive protocols where we wanted to solve a bigger problem rather thanthe problem of secure communication. So in this lecture we will introduce Zero-knowledge protocols. We will see some of the motivation first in the zero-knowledge protocols and a formal definition. And we will see a zero-knowledge protocol for the graph isomorphism problem.

(Refer Slide Time: 00:52)
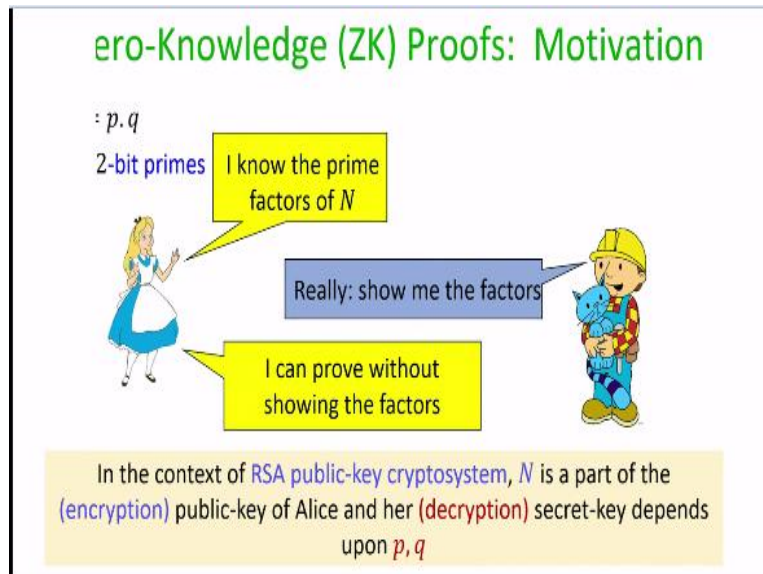
Now let us start by trying to understand a motivation for zero-knowledge proofs. So imagine we have two parties Alice and Bob and say Alice had picked two random prime numbers p and q say each of size 512 bits and it has computed the product of p and q to obtain the product N and she goes and claims to Bob that I know the prime factors of N. Now if indeed she wants to prove to Bob that she knows that prime factors of N then a simple way to prove that is to show the values of p and q.

Because if the values of p and q are given to Bob, Bob itself can multiply those two numbers and see whether its matches N or not. But that is was we call as proof in clear. Because p and q are here witness for Alice for the statement that she knows the prime factors of N. One way of proving her statement is to show the witnesses in clear but what a zero-knowledge protocol is going to do here is its going to allow Alice to convince Bob that indeed she knows the prime factors of N without actually showing the witnesses namely p, q.

Because p and comma q might come secret information for Alice and in this whole process of zero-knowledge proof basically Alice ends up convincing Bob that she knows p and q without actually revealing p and q. Now you might be wondering where this p and q is useful. So if you remember in the context of RSA public-key cryptosystem, the value N is nothing but a part of the public-key of Alice and p, q are nothing but part of a decryption key.

So if indeed Alice want to convince to Bob that N is her public key and she knows the corresponding secret key without showing the components related to the secret key namely p and q; she we can convenience Alice by using this zero-knowledge proof.

(Refer Slide Time: 02:45)
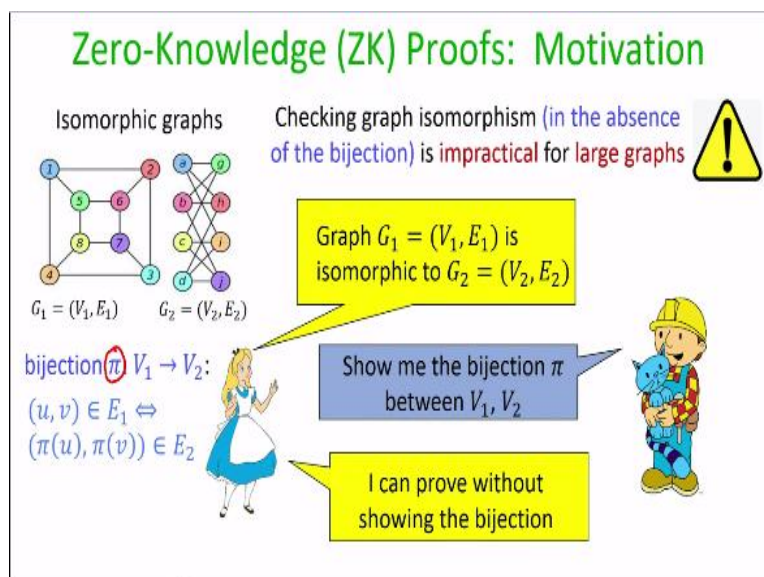
Let us see another motivation. Imagine Alice has two graphs here. And pictorially these two graphs are drawn in a different way. The vertex names are different; the edge names are different then so on. But it turns out that information wise the two graphs are isomorphic or structurally the graphs are isomorphic to each other and what I mean by isomorphic graphs here is that, if you consider these two graphs then for these two graphs there exist a bijection from the vertex set of the first graph to the vertex set of the second graph.

And that bijection has the property that if there is an edge between the nodes u and v in the first graph then in the second graph there exists an edge between the mapped u set and mapped v set. And this is an if and only if statement. That means in the same way if you have an edge between the mapped u vertex and the mapped v vertex in the second graph then an edge exists between the u vertex and the v vertex in the first graph. So in that sense these two graphs are isomorphic even though pictorially they are looking different.

So imagine Alice has two isomorphic graphs, so she makes the description of the two graphs available to Bob and she makes the statement that she claims to Bob that these two graphs are

isomorphic. So that is the statement. Again Bob cannot directly check whether the two graphs are isomorphic or not and verify Alice claim because its impractical to verify whether two graphs are isomorphic or not in the absence of the bijection which is available or which maps the vertex set of the first graph to the vertex set of the second graph.

So one way for Bob to verify the statement of Alice is it can ask Alice that please give me your witness namely the bijection Π. If you give me the bijection Π then I can verify whether indeed for every edge in the graph, first graph the corresponding mapped vertices also constitute an edge in the second graph and so on. But that will be a proof in clear. So what a zero-knowledge proof or a zero-knowledge protocol will allow Alice to do is; it will allow Alice to convince Bob that indeed these two graphs are isomorphic without actually showing the witness namely the bijection Π.

(Refer Slide Time: 05:10)
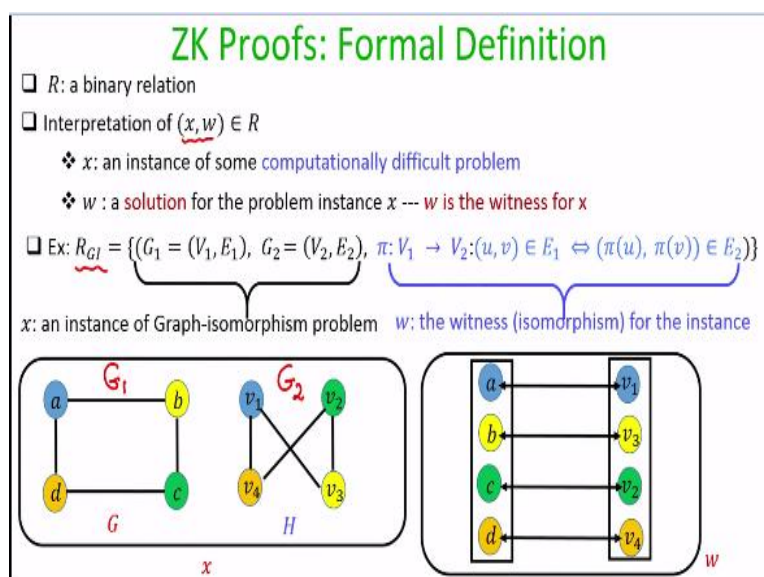
So that means a zero-knowledge proof is a kind of an interactive protocol between two entities a prover and a verifier which allows a prover to prove a statement to verify without actually showing anything about the underlying witness. So now let us formalize this statement. So imagine R is a binary relation and the interpretation of this relation is as follows. So if I have a pair (x, w) present in this relation R that should be interpreted as if x is an instance of some computationally difficult problem.

When I say computationally difficult problem informally it means in poly amount of time it is not known how to solve that problem. But again that is a very loose statement but that is an intuitive understanding of computationally difficult problem here and a w here is a solution for that problem instance x namely you can consider the w to be a witness for the problem instance x. So to understand this relation R let us consider the graph isomorphism relation here.

So an (x, w) pair in the graph isomorphic relation R_{GI} will look like this {(G$_1$ = (V$_1$, E$_1$), (G$_2$ = (V2, E2))} and the interpretation of elements present in this graph isomorphism relation should be as follows. So the first part here is the description of the two graphs namely it is a problem instance. That means someone wants to prove or disprove these two graphs are isomorphic. And the corresponding witness is nothing but the isomorphic or the bijection from the vertex set of the first graph to the vertex set of the second graph.

So for instance if you consider these two graphs here these two graphs are indeed isomorphic. So first graph is your graph G$_1$ and the second graph here is your graph G$_2$. So that is the x component of the any x, w which might be present in this graph isomorphism relation. And a corresponding witness with respect to this specific G$_1$ G$_2$ graph is nothing but the bijection from the vertex set of graph G$_1$ to the vertex set of graph G$_2$ and so on.
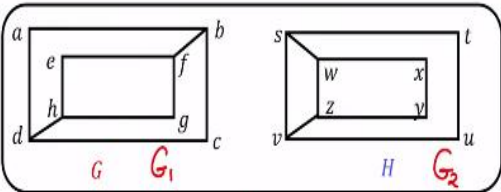
(Refer Slide Time: 07:18)

ZK Proofs: Formal Definition

- R: a binary relation
- Interpretation of $(x, w) \in R$
  - x: an instance of some computationally difficult problem
  - w : a solution for the problem instance x --- w is the witness for x
- Ex: $R_{GI} = \{(G_1 = (V_1, E_1),\ G_2 = (V_2, E_2),\ \pi: V_1 \rightarrow V_2 : (u, v) \in E_1 \Leftrightarrow (\pi(u), \pi(v)) \in E_2)\}$

x: an instance of Graph-isomorphism problem w: the witness (isomorphism) for the instance

No witness for this instance x (as the graphs are non-isomorphic) and hence does not belong to R_{GI}

On the other hand, if I consider these two graphs to be my G_1 and G_2, it turns out that these two graphs are not isomorphic to each other and as a result if this is my candidate x then for this candidate x I do not have any candidate w such that x, w belongs to this relation R_{GI}, right. That means only those x, w will be present in r where the x instance has a corresponding witness w. If for an instance x there exists no witness w such that x, w satisfies that relationship then we say that, (x, w) is not present in the relation R.

(Refer Slide Time: 07:57)
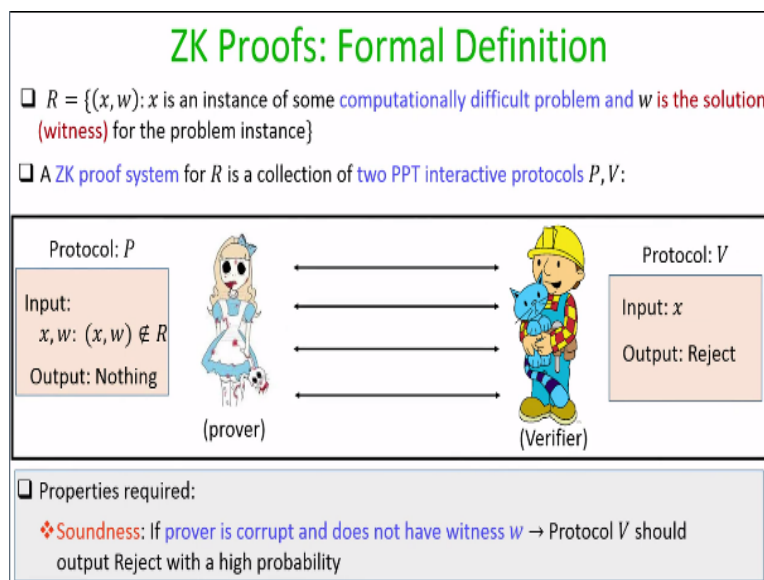
So now let us go into the formal definition of zero-knowledge proofs. So we are given some publicly known relation which will have elements of the form x, w where x is an instance of some computationally difficult problem and w is the solution or the witness for that instance. Then a zero-knowledge proof system for the relation R is consist of two poly time algorithm two randomized algorithms one for the prover and one for the verifier.

The input for the prover will be an x, w pair and the goal of the prover is to prove it knows a w such that x, w belongs to r whereas the input for the verifier algorithm will be the problem instance x. And the goal of the verifier is to verify whether indeed Alice knows a w such that x, w belongs to R or not. So in the zero-knowledge proof system the prover will send messages or it will interact with the verifier where the messages for the prover will be computed as per the protocol P and the internal randomness which are used by the prover.

And at the end of the protocol prover outputs nothing whereas the verifier algorithm it will interact with the prover where the messages of the verifier will be computed as per the algorithm V and the internal randomness chosen by the verifier and the verifier is either going to output accept or reject. Accept means it accepts the fact that Alice knows some witness, reject means it does not believe in Alice statement.
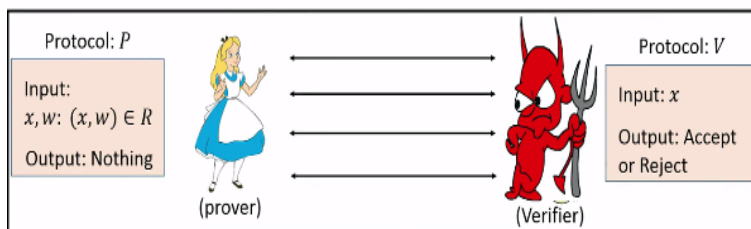
Now what are the properties we require from a zero-knowledge proof system, the first property is the completeness property which demands that if both prover and verifier are honest and if indeed prover knows an x, w such that x, w belongs to R and both prover and verifier participates, performs all their actions as per the protocol P and V. Then with a very high probability the output of the verifier should be accept.

That means verifier should accept the statement of prover. The second property is Soundness property which we consider with respect to a corrupt prover. So what we require here is that if my prover is corrupt and it does not have an x, w such that x, w belongs to the relation R then irrespective of however she participates in the protocol the output of the verifier should be reject. That means a malicious prover who does not have x, w or who does not have a witness such that x, w belongs to the relation R with very less probability the prover should be able to convince the verifier that she knows the witness w. That means with very high probability the verifier should be able to catch a malicious prover. That is the soundness requirement.

(Refer Slide Time: 10:50)

ZK Proofs: Formal Definition

❑ $R = \{(x, w): x$ is an instance of some computationally difficult problem and w is the solution (witness) for the problem instance$\}$

❑ A ZK proof system for R is a collection of two PPT interactive protocols P, V:

Protocol: P

Input: $x, w: (x, w) \in R$

Output: Nothing

(prover)

Protocol: V

Input: x

Output: Accept or Reject

(Verifier)

❑ Properties required:

❖ Zero-knowledge: If prover is honest and verifer is corrupted → probability distribution of the values received by the verifier is "independent" of w

And the third property is a zero-knowledge property which is with respect to a corrupt verifier and a honest prover. So zero-knowledge property demands that if the prover is honest and the verifier is corrupt then irrespective of the way the verifier algorithm or the verifier participates in this protocol the verifier learns absolutely nothing about the witness w that is available with the prover. So here nothing is in-quote, unquote "nothing". It is not formal.

What exactly we mean by nothing is learned about the witness. If you want to little bit more formal we can say that, we say that the protocol has the zero-knowledge property if the probability distribution of the transcripts seen by the verifier is independent of the actual witness which is available with the prover. Again I am not formally proving what exactly it means to say that the probability distribution of the transcripts seen by the verifier is independent of w.

The actual formalism is little bit subtle and involved. So let us not go into the actual detail but for your understanding you can imagine that verifier should not learn anything about the witness if the verifier is corrupt by participating in this protocol.

(Refer Slide Time: 12:06)
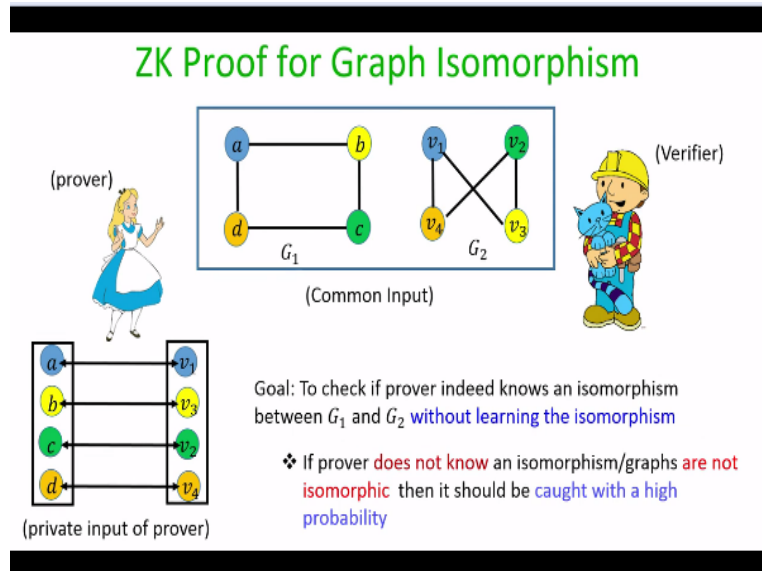
So now let us see a zero-knowledge proof system for a graph isomorphism problem. So the common input that is available to both prover and a verifier is an instance of a graph isomorphism problem namely the description of two graphs. And Alice wants to convince Bob or the verifier here the prover; the Alice is prover and the Bob is verifier. So Alice wants to prove to Bob that indeed these two graphs are isomorphic.

And she claims that she knows the isomorphism between these two graphs, that means she claims that she has a private input mapping the vertex set of the first graph to the vertex set of the second graph with respect to which these two graphs are isomorphic and the goal here is to design a protocol which allows Bob to learn whether indeed these two graphs are isomorphic or not without actually learning the isomorphism.

And it should be sound in the sense that if prover does not know the isomorphism or the graphs are not isomorphic at the first place then she should be caught while giving the proof with a very high probability.

(Refer Slide Time: 13:07)

ZK Proof for Graph Isomorphism
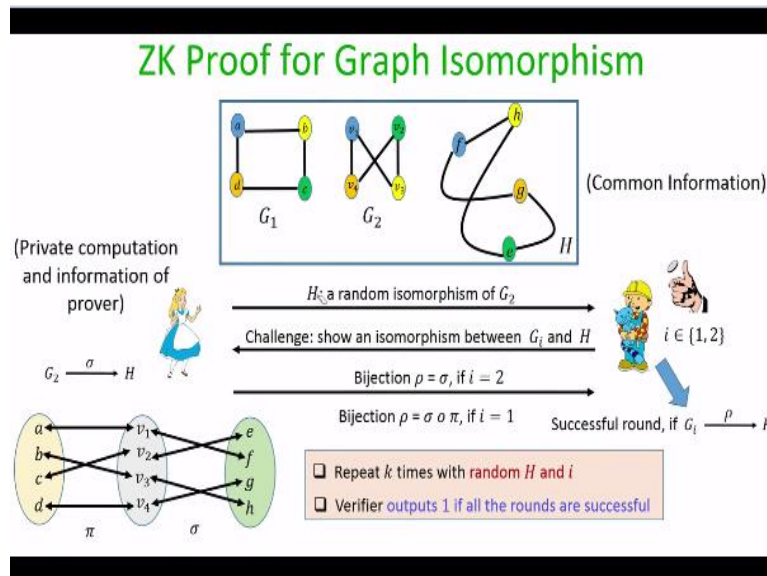
So let us see how exactly the zero-knowledge protocol is designed here. So this rectangle box I have highlighted the public information so the public information available both to Alice and Bob are the description of the two graphs. And the private information available with Alice, the prover is the witness Π or the mapping from the vertex of first graph to the vertex set of the second graph.

So the first round of the zero-knowledge protocol is as follows. So what Alice does is, she creates a random isomorphic copy of the second graph G_2. And that is very easy to do, what she has to do is she has to basically come up with a random permutation say σ mapping the vertex set of the second graph and the resultant mapped vertices is going to give her another graph H.

So once she computes a random isomorphic copy of the second graph she sends the description of the random isomorphic copy of the second graph to Bob. So that is kind of a commitment. So I stress σ is randomly chosen and known only to Alice here. Now what Bob does is, Bob picks a random coin and with probably 1/2 the random coin could give the output 1 or with probability 1/2 it could give the output 2.

And now what Bob's challenge is, Bob challenges Alice to show an isomorphism between the ith graph and the new graph H which Alice has committed. So if i = 1 then basically Bob is challenging Alice to show an isomorphism between graph G_1 and the new graph H whereas if i =

2 then Bob is challenging Alice to show an isomorphism between the graph G_2 and the new graph H. I stress here that when Alice was committing the graph H, she does not know well in advance whether she will be challenged to show an isomorphism between G_1 and H or between G_2 and H. She does not know that in advance.

From here viewpoint with probability 1/2 she could be challenged to show an isomorphism between G_1 and H and with probability 1/2 she could be challenged to show an isomorphism between G_2 and H. Now once Bob throws the challenge Alice has to respond. And the response will be different depending upon whether i = 1 or whether i = 2. Namely if the challenge is i = 2 that means if Alice is challenged to show the isomorphism between G_2 and H then she can simply supply the bijection σ which she has used to compute H from the graph G_2 graph.

So that will be her response. So I am denoting the response by ρ here. So ρ will be; the response ρ will be nothing but the mapping σ if i = 2. On the other hand, if Bob has challenged Alice to show an isomorphism between graph G_1 and H, then basically Alice can respond back by composing the mapping Π which was her witness with the secret mapping σ that he has chosen to go from the graph G_2 to H.

Because if we compose Π and σ then by using Π from G_1 we come to G_2, Alice comes from G_1 to G_2 and then composing again with the mapping σ from G_2 she can go to the graph H. So that means the way to go from G_1 to H is to compose the mapping Π with the mapping σ. And if indeed Alice knows Π she should be able to compose Π and σ and she can respond back with this response ρ.

Now Bob has to verify whether indeed Alice has properly responded or not. So what Bob checks is it knows that Alice is suppose to show an isomorphism between the ith graph and H and the response from Alice is mapping ρ and Bob just has to verify whether indeed the graph G_i takes Bob to the graph H as per this mapping ρ or not. If it is then Bob is convinced that Alice has successfully passed this round otherwise Bob says that Alice has failed in this round.

And what Bob is going to do is Bob is going to repeat this whole process namely Alice committing something Bob challenging and Alice again submitting the response k number of times and for each of these rounds Alice has to pick random commitment H that means she will be freshly picking the graph H for every round or every iteration and in the same way Bob will be randomly picking the challenges i for every round, independent of every previous round. That means it will not be the case that in every round Bob will be picking i = 1 or i = 2.

And the verifier Bob is going to output 1 namely it will say that indeed Alice knows the secret mapping Π, if all the rounds are successful from the viewpoint of Bob, that means in all the rounds Alice has successfully submitted the responses ρ. Whereas if any of the rounds Alice fails then Bob outputs reject. That means Bob rejects the claim of Alice. That is the zero-knowledge protocol here for the graph isomorphism problem.

(Refer Slide Time: 18:47)
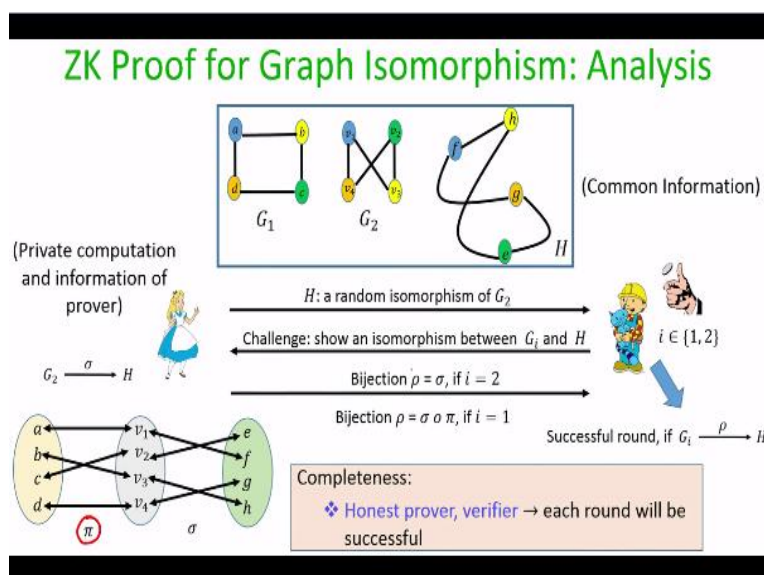
Now let us do the analysis, let us see whether this graph isomorphism protocol satisfies the requirements of correctness, soundness and zero-knowledge or completeness, soundness or zero-knowledge. So the completeness property or the correctness property here is straightforward, if indeed Alice is honest that means she knows the secret mapping Π between the graph G_1 and G_2 and if she is following the protocol instructions honestly and if verifier is also honest then each of the round will be successful. Because it does not know whether Bob challenges with i = 1 or

with i = 2. Alice will always be able to successfully respond with the right mapping ρ and hence all the rounds will be successful.

(Refer Slide Time: 19:30)
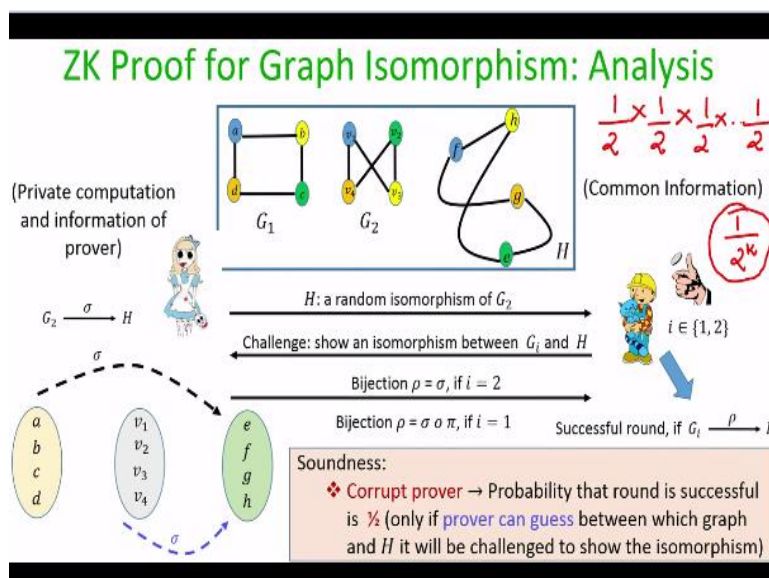
Now let us analyze the soundness property here and recall for the soundness property we have to consider the case when Alice is potentially corrupt. And she does not know the mapping between G_1 and G_2 or at the first place there may not be any mapping between G_1 and G_2 showing that they are isomorphic and so since she is corrupted she might not follow the protocol, and remember as per the protocol she is supposed to send an isomorphic copy of G_2 in every round but she not do that as well.

So for soundness we have to analyze that with how much probability she can successfully cheat even though she does not follow the protocol and she does not have the isomorphism between the graph G_1 and G_2 available with her. It turns out that the only way she can cheat in a round is to guess in advance what will be the challenge from the honest Bob. Because if she can guess correctly well in advance whether i = 1 or whether i = 2 then at the first place itself she can create an isomorphic copy of G_i.

So recall that as per the protocol steps she supposed to create an isomorphic copy of G_2 namely H should be an isomorphic copy of G_2. But if Alice is corrupt she may not be able to follow the protocol, she may try to guess in advance that I could be 1, I could be 2 and with respect to that

guess she can create an isomorphic of that graph G_i. And if indeed her guess is correct she will be able to successfully submit the response ρ.

But the probability that she can correctly guess whether i is going to be 1 or whether i is going to be 2 is 1/2. That means with only probability 1/2 she can cheat in one round and end up successfully passing that round. But since we are repeating this process k number of times, right because remember Bob is not convinced just by doing this protocol once; depending upon how much confidence he want to be in the claim of Alice, he may repeat the protocol k number of times.

So what is the probability that in each of the k iteration a bad Alice who does not know the isomorphism between G_1 and G_2 successfully end up passing all the k rounds. The probability that she is successful in the first round is 1/2; the probability that she successful in the second around is also 1/2. And remember, the probability of success; getting successful in the second round is independent of the probability of getting successful in the first round.

Because in the second round also the challenge of Bob will be independent of the challenges that Bob has picked in the first round. So that is why the probability that Alice is successful in the second round is 1/2 and like that the probability that Alice is successfully able to cheat Bob in all the rounds by correctly guessing in advance the challenge of Bob in all the rounds is 1/2 * 1/2 * 1/2 * 1/2 … k times which is nothing but $1/2^k$.

So if k is significantly large say imagine k = 100 and if a Alice does not know the isomorphism between G_1 and G_2 then definitely there exists at least one of the rounds with very high probability where Alice will be caught and hence Bob will reject the statement of Alice. The only case Alice will be successful in all the rounds is when she is able, when she is lucky enough to guess the challenge of Bob in all the k rounds in advance which can happen only with probability $1/2^k$ which is very small if k become significantly large.

(Refer Slide Time: 23:11)

ZK Proof for Graph Isomorphism: Analysis
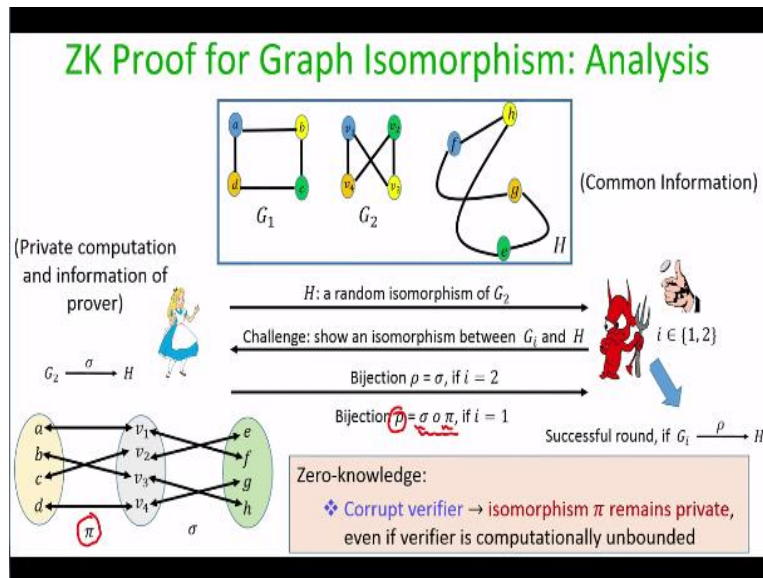
Now let us try to understand the zero-knowledge property here where; remember for zero-knowledge we have to consider the case where Alice is honest and Bob is corrupt, the verifier is corrupt. And the goal of the corrupt verifier is to analyze the protocol transcript and learn the secret permutation Π which maps the graph G_1 to graph G_2 namely the isomorphism between graph G_1 and G_2.

It turns out that in each round the corrupt verifier does not learn anything about the mapping Π, because if the challenge from Bob is $i = 2$ then the response that Alice throws is the mapping from the graph G_2 to H which does not reveal anything about a secret mapping Π. On the other hand, if the verifier challenges Alice with $i = 1$ then in that case Alice responds with the composition of the secret mapping Π with another randomly chosen mapping σ.

And since σ is randomly chosen; in the iteration you can imagine that σ is acting as some kind of mask here, because since σ is randomly chosen and Π is any how randomly chosen and available only with Alice; this overall composed mapping σ does not reveal anything about the secret mapping Π because the masking here namely the secret mapping σ is randomly chosen by Alice.

And since σ is randomly chosen in each iteration independent of all the iterations even if a malicious prover keeps on challenging Alice with $i = 1$ it will fail to learn about the secret mapping Π, and this holds even if the verifier is computationally unbounded. So in that sense a

malicious Bob does not learn anything about the secret witness Π available with Alice. So that shows that this zero-knowledge proof system that we have designed for the graph isomorphism problem indeed satisfies all the requirements of a zero-knowledge proof system for a graph isomorphism problem.

So that brings me to the end of this lecture. Just to summarize in this lecture we have introduced the problem of zero-knowledge proof system, we have formally stated their requirements namely the completeness, soundness and zero-knowledge requirement and we have also seen an instance of the zero-knowledge proof system for the graph isomorphism problem. Thank you.