CSA E0 312: Secure Computation October 30, 2015

Guest Lecture 4

Guest Instructor: C. Pandu Rangan *Submitted by: Cressida Hamlet*

1 Introduction

Last lecture, we saw what ZKP is and why we need it. We also saw sigma protocol, a usual method for proving in zero knowledge. Is sigma protocol sufficient to prove everything? Let's first look into sigma protocol. Firstly prover should have a witness or a proof of something to be proved, and the prover give a commit to the verifier. Now verifier sends a challenge and according to the challenge, prover sends response. Verifier will check this transcript and accept or reject prover. What happens if there is no witness? How will prover commit something with out having a particular witness?

2 Proof of Assertion

To understand the case where there is no witness, let us look into the following two problems.

First problem is to prove 2 graphs G_1 and G_2 are isomorphic. If two graphs are isomorphic, then there exists some permutation π which is one to one and onto mapping of vertices of G_1 to vertices of G_2, such that if there is an edge between two vertices in G_1, then there will be an edge between the corresponding vertices in G_2. Here π can act as the witness the prover can use to prove G_1 and G_2 are isomorphic.

Second problem is to prove 2 graphs are non-isomorphic, which means there exist no such permutation π between the verices of G_1 and G_2. So in this case, even if prover has some mechanism to find out the graphs are non-isomorphic, he/she doesn't possess any witness. Here prover needs to prove the assertion that the graphs are non-isomorphic - *Proof of Assertion*. In this case, prover won't be able to initiate the proof as in sigma protocol.

Graph Isomorphism	**Graph Non-Isomorphism**
Exists a witness	Doesn't exist a witness
Prover has to convince the possesion of the witness	Prover has some assertion which needs to prove
Proof of Knowledge	Proof of Assertion
Can use Sigma protocol	Can't use Sigma protocol

We now know that we can't use sigma protocol to prove if there exists no witness. So let's see an example for (zero knowledge) proving something which doesn't have a witness.

2.1 Colour Blind Verifier

Consider the case where verifier, who is colour blind, has 2 balls of same size and prover needs to prove that those are of different colours. In this case, there exists no witness to

prove this, but prover has some method (say prover is not colour blind and so can recognise colours) to differentiate the balls which looks exactly the same except for its colour. Now how can the prover prove that the balls are of different colour?

Verifier can show one ball at a time to the prover and prover can say which colour the ball has, and next time, verifier can show either of the balls randomly and let prover say that if it is the same ball or the other ball. For a honest prover, he/she can always distinguish between two balls and for a dishonest prover even though can't distinguish between the balls, will be able to guess with a probability $1/2$. So if this is repeated k times then the probability of cheating by a dishonest prover is just $1/2^k$.

Now if we look into the above protocol, we can see that there was no witness which is used to prove this or there was no commitment phase. The protocol was initiated by the verifier (not prover as in sigma protocol). Also this protocol doesn't give any new knowledge to the verifier except that the assertion is correct. Now let us come back to the graph non-isomorphic problem.

2.2 Graph Non-Isomorphism

We now know that verifier starts the protocol. Verifier has 2 graphs, G_1 and G_2, and prover needs to prove that he/she has some mechanism to test whether 2 graphs are non-isomorphic and thus know that these 2 graphs are non-isomorphic. One round of the protocol is given below.

Verifier:

- Picks a random i from $\{0, 1\}$ and a random permutation (π) for the vertices of G_i
- Let $\mathsf{H} = \pi(\mathsf{G})$. Send H to prover.

Prover:

- Check if which of G_1 or G_2 is isomorphic to H.
- If G_i is isomorphic to H, send i as the response to the verifier.

Now if r is the response, verifier checks if $r = i$ and if not same, rejects the prover, if same then repeat this for k times to reduce the error probability to $1/2^k$.

Zero knowledge is obivous, so let's see if this protocol satisfies the soundness and completeness. Consider the following cases:

- If the prover is honest:
  - G_1 and G_2 are non-isomorphic
  - Prover has some mechanism to check if G_i is isomorphic to H
  - Which means, prover will always get accepted.
- If the prover is dishonest
  - If G_1 and G_2 are not non-isomorphic:
    - Only way for prover to find i is a random guess, since G_1 and G_2 are isomorphic means H is isomorphic to both.

* Which gives the probability of getting accepted in one round is 1/2. So if there are k rounds, the the error probability (prover always guess i correctly) is $1/2^k$.

– If G_1 and G_2 are non-isomorphic, but prover doesn't know how to check it:

* Here, as in the previous case, prover can't guess i correctly with probability more than 1/2, which means if there are k rounds, then the error probability is $1/2^k$.

3 Proof of Knowledge

Now let us see some example protocols for zero knowledge proof of knowledge.

3.1 Graph Isomorphism

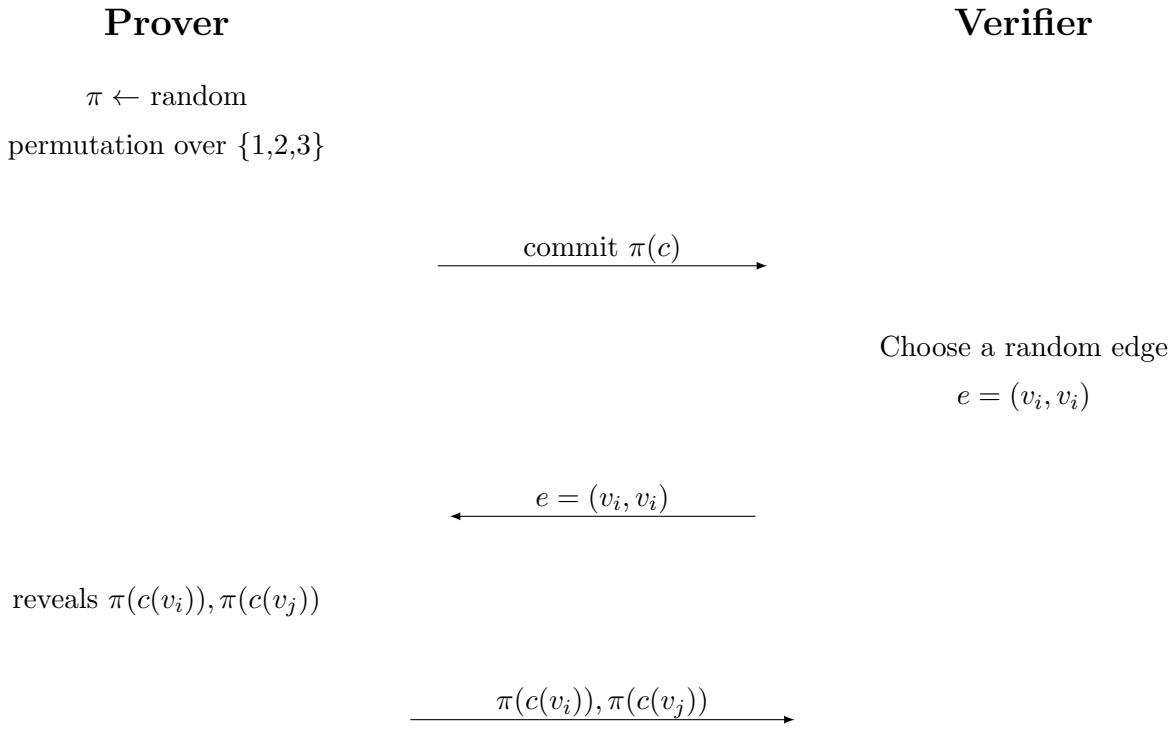
In graph isomorphism, verifier has two graphs G_1 and G_2, and prover needs to prove that these graphs are isomorphic and he/she has the permutation π, which as we said before. Here, since a honest prover will always have a witness, we can use sigma protocol for proving. Now let us see the zero knowledge proof protocol for this:

Prover		**Verifier**
Choose a random permutation π_R $\mathsf{H} = \pi_R(\mathsf{G})$		
	$\xrightarrow{H}$	
		Choose a random number $s \in \{1, 2\}$
	$\xleftarrow{s}$	
If $s == 2$, $\pi_{resp} = \pi_R \circ \pi$ else $\pi_{resp} = \pi_R$		
	$\xrightarrow{\pi_{resp}}$	
		check if $\pi_{resp}(\mathsf{G}_s) = H$ if true accept, else reject

It is obivous that the above protocol is zero knowledge since, verifier either gets a random permutation or π composed with a random permutation. In both cases, verifier doesn't gain any new information regarding π. We can also see that a honest prover will always gets accepted and a dishonest prover will get accepted only with a probability $1/2^k$, where k is the number of rounds, the protocol is repeated (has the same explanation as the above problem).

3.2 3-Colouring of Graph

In 3-colourability, we check if a graph is 3-colourable. A graph is 3-colourable means, there is a mapping c from vertex set to a set of 3 colours, such that, for any edge, end points will have different colours. Verifier has a graph G and prover needs to prove that the graph is 3-colourable and he/she has the colouring function c.



The commiting needs two properties:

- Binding: The commiting of π binds π means, once if it is commited, then at the time of revealing prover shouldn't be able to show some other value other than what he/she is commited. So if prover commits to some colours for each vertex, after getting chanllege (a random edge), prover can only reveals the same colours for that edge. Binding is useful to not get cheated by a dishonest prover. Since we doesn't have any assumption for the computational powers of prover, we need perfect binding in this protocol.
- Hiding: Commiting hides means, from the commit one won't get any new information

about what is there commited unless it is revealed. Hiding helps not to leak information to a dishonest verifier, who tries to get extra knowledge from the interaction. Since we assume verifier is polynomial bounded, computational hiding is sufficient.

Now let us check if this protocol satisfies all the properties of ZKP. Since there is computational hiding and we assume that the verifier is computationaly bounded, verifier won't be able to get any new knowledge from the interaction and so this satisfies zero knowledge property.

A honest prover can always commit a colouring such that each edge has different colours and so a honest prover will always gets accepted.

Since perfect binding property is satisfied by the commiting scheme, prover can't commit to one colour for a vertex and later change it. So if prover doesn't know 3-colouring of the graph, then there will exists atleast one edge with same colours and with probability $1/|E|$, verifier will pick this edge as challenge and prover will get caught. So if we repeat this $n|E|$ times, then the error probability is

$$\left(1 - \frac{1}{|E|}\right)^{n|E|} \sim e^n$$

4 Summary

Now we have seen that there are mainly 2 types for ZKP, where there exists witness and where there doesn't exists witness. When it is to prove a yes instance of a decision problem, usually there exists a witness and so we will be able to use sigma protocol for ZKP. If it is a no instance, usually there won't be a witness and verifier will need to initiate the protocol. We have also seen some examples for ZKP in both cases in this lecture.