CSE 594 : Modern Cryptography 04/11/2017

Lecture 19: Zero-Knowledge Proofs - II

Instructor: Prof. Omkant Pandey *Scribe: Swarnima Shrivastava, Vaishali Chanana*

1 Recall: Zero Knowledge and Interactive Proof for Graph Isomorphism

In last class, we proved completeness of interactive proofs i.e., a prover can't cheat. We discussed soundness of the proof but we did not prove that the proof itself doesn't convey any extra knowledge. In today's class, we will prove that the interactive proof itself is a zero knowledge proof. Following is the definition:

Definition 1 *An interactive proof (P, V) for a language L with witness relation R is said to be zeroknowledge if for every non-uniform PPT adversary V^*, there exists a PPT simulator S s.t. for every non-uniform PPT distinguisher D, there exists a negligible function $v(.)$ s.t. for every $x \in L$, $w \in R(x)$, $z \in 0, 1^*$, D distinguishes between the following distributions with probability at most $v(|x|)$:*

- $\{View^*_V[P(x, w) \leftrightarrow V^*(x, z)]\}$
- $\{S(1^n, x, z)\}$

Note: If the distributions are statistically close, then we call it *statistical zero knowledge* and if the distributions are identical, then we call it *perfect zero knowledge*

Now, let us prove that the interactive proof (Graph Isomorphism) is a perfect zero knowledge proof i.e., it doesn't convey any extra knowledge. Recall Graph Isomorphism protocol from last class:

Protocol (P,V): Prover P, Verifier V, common input $x = (G_0, G_1)$ and $P's$ witness π s.t. $G_1 = \pi(G_0)$, repeats the following procedure n times using fresh randomness each time:

- $P \rightarrow V$: Prover chooses a random permutation $\sigma \in \Pi_n$, computes $H = \sigma(G_0)$ and sends H
- $V \rightarrow P$: V chooses a random bit $b \in 0, 1$ and sends it to P.
- $P \rightarrow V$: If $b = 0$, P sends σ, else $\phi = \sigma.\pi^{-1}$
- V outputs 1 iff $H = \phi(G_b)$ where verifier $V's$ view is $V(x, b, \phi)$

This way the verifier is never able to see π. He either sees σ or ϕ and since σ is completely random, ϕ is also random and is computationally indistinguishable even though it was calculated using π. That's the protocol and it is sound because if the graphs are not isomorphic then with a prover can guess correct answer only with probability 1/2.

2 Formal Proof: Protocol(P,V) is Perfect Zero Knowledge

Strategy: We are going to prove that a single iteration of (P, V) is perfect zero knowledge and the way we have defined zero knowledge, the following theorem holds which states that if we can prove the property for one component then it holds for all the subsequent sequential executions.

Theorem 1 *Sequential repetition of any zero knowledge protocol is also zero knowledge.*

Now, all we need to do is prove that single iteration of $Protocol(P, V)$ is perfect zero knowledge. To prove it, we need to do the following:

1. Construct a Simulator S for every PPT V^* s.t. the view of verifier V in real execution is indistinguishable with view created by simulator.
2. Prove that expected run time of such a simulator S is polynomial
3. Prove that the output distribution of S is correct (i.e.indistinguishable from real execution)

2.1 Construction of Simulator S

Main idea behind constructing such a simulator is if the simulator is not able to produce results similar to real execution, it throws away the information about current run and restarts afresh till it produces a run computationally indistinguishable to real execution.

Algorithm 1 Simulator $S(x, z)$:

1: Choose random $b' \xleftarrow{\$} \{0,1\}$ and $\sigma \xleftarrow{\$} \Pi_n$
2: Compute $H = \sigma(G_{b'})$
3: Emulate execution of $V^*(x, z)$ by feeding it H. Let b denote its response.
4: If $b = b'$, then feed σ to V^* and output its view. Otherwise, restart the above procedure.

In above algorithm, we assumed that G_0 and G_1 are isomorphic. Since G_0 is isomorphic to G_1, for a random $\sigma \leftarrow \Pi_n$, $\sigma(G_0)$ and $\sigma(G_1)$ are identically distributed i.e., distribution of H is independent of b'. Therefore, H has the same distribution as $\sigma(G_0)$. Now, since V^* only takes H as input, its output b' is also independent of b'. Since b' is chosen at random, a cheating verifier can only guess it with probability no more than $half$. More formally,

Lemma 2 *In the execution of $S(x, z)$,*

- *H is identically distributed to $\sigma(G_0)$, and*
- $Pr[b = b'] = \frac{1}{2}$

2.2 Proving Run Time of Simulator S is Polynomial

Some relaxation is provided here in terms that instead of asking for a strict polynomial time algorithm, we can consider a running time that is polynomial only in expectation i.e., run time itself might be random whose expectation is polynomial since this can be addressed.
Claim: If X is a random boolean variable s.t. $Pr[X = 1] = p$ and T be random num of trials before $X = 1$ occurs then, $E[T]=\frac{1}{p}$ provided $0 < p < 1$.

Now from above lemma, we know that S can succeed with probability $\frac{1}{2}$ in each trial. Therefore, in expectation, S stops after 2 trials. Since each trial takes polynomial time, so total run time of S is expected to be polynomial.

2.3 Proving Indistinguishability of Simulated View

From above lemma, we know that H has same distribution as $\sigma(G_0)$. If we could always output σ, then output distribution of S would be same as in real execution. S, however, only outputs H and σ if $b^{'} = b$. But since H is independent of $b^{'}$, this does not change the output distribution. Therefore, we can say that the simulated view is computationally indistinguishable from view obtained from real execution.

2.4 Takeaway:

A verifier V that participates in the live conversation with the prover P is convinced that transcript x is valid but any 3^{rd} party (verifier) who did not see the live conversation with the prover could not be convinced that the transcript is real/valid since it is easy to come up with a simulated transcript without knowing the actual isomorphism. It is similar to signature authentication only if it is done online. If you get the transcript from third party, it might not be authentic as it could be forged using simulator. This is the main essence, tt gives you plausible deniability.

3 Zero-Knowledge Proofs for NP Languages and Graph 3-Coloring

If we construct proofs for every NP language, it would not be efficient. There are numerous languages in NP. Key to this is NP-completeness. All languages in NP reduces to NP-complete languages. Also, the following theorem states as follows:

Theorem 3 *If one-way permutations exist, then every language in NP has a zero-knowledge interactive proof.*

This assumption can also be relaxed to just one-way functions.
Using this fact, we can say that instead of constructing proofs for all NP languages, it would suffice if we construct a zero knowledge proof for an NP-complete problem(say $Graph3-Coloring$ in our case) and reduce NP problem to it as shown below:

1. Given instance x and witness w, P and V reduce x into an instance $x^{'}$ of Graph 3-coloring using Cooks (deterministic) reduction
2. P also applies the reduction to witness w to obtain witness $w^{'}$ for $x^{'}$
3. Now, P and V can run the zero knowledge proof from Step 1 on common input $x^{'}$

3.1 Physical Zero knowledge Proof for Graph 3-Coloring

Consider a graph $G = (V, E)$. Let C be a set of 3-coloring of V given to P. P then picks a random permutation π over colors $\{1, 2, 3\}$, colors G according to $\pi(C)$ and hides each vertex inside a locked box(we will digitize this later). Conceptually, using locked box is a way of saying that V doesn't have access to know the color of any of the vertices initially. Now, once V picks a random edge

$(u, v)$from E, P unlocks the boxes corresponding to edge (u, v). V accepts if u and v have different colors, and rejects otherwise. The above process is repeated $n|E|$ (to ensure simulator's success, as discussed in section 2.2 above)

Intuition behind zero knowledge proof of Graph 3-coloring:

1. Intuition for Soundness: In each iteration, cheating prover is caught with probability $\frac{1}{|E|}$
2. Intuition for Zero Knowledge: In each iteration, V only sees something it knew before two random (but different) colors.

3.2 Digitizing above Proof

To digitize the above proof(intuitions), we need to implement our concept of *lockedboxes*. We require the digital lock boxes to hold the following two properties:

1. **Hiding**: V should not be able to see the content inside a locked box
2. **Binding**: P should not be able to modify the content inside a box once its locked.

This can be achieved with the help of technique called *commitmentschemes* which is discussed in the following section.

4 Commitment Schemes

4.1 Introduction

The locked boxes described above can be well explained by their digital analogue - *commitment schemes*. If we have one-way permutation, we will be using the non-interactive version of the scheme and if we do not have, then we might need a few rounds of interaction. It has two phases:

1. **Commit phase**: Sender locks a value v inside a box and sends it over to the verifier.
2. **Open phase**: Sender unlocks the box and reveals v.

Definition 2 *A randomized polynomial time algorithm Com is called a commitment scheme for n-bit strings if it satisfies the following properties:*

- ***Binding**: For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$, it holds that $Com(v_0; r_0) \neq Com(v_1; r_1)$*
- ***Hiding**: For every non-uniform PPT distinguisher D, there exists a negligible function $v(.)$ s.t. for every $v_0, v_1 \in \{0,1\}^n$, D distinguishes between the following distributions with probability at most $v(n)$*
  - $\{r \xleftarrow{\$} \{0,1\}^n : Com(v_0; r)\}$
  - $\{r \xleftarrow{\$} \{0,1\}^n : Com(v_1; r)\}$

The definition is applicable to a single message. As we are coloring for many nodes, we need the hiding property for multiple messages. Intuitively, instead of a single v_0, we can have sequence of v in the distribution and they should look indistinguishable. This is called *multi-value hiding* which is explained in the next section.

4.2 Construction of Bit Commitments

Let f be an OWP and h be the hard core predicate for f. To commit bit b, you chose random permutation r such that sender computes $Com(b;r) = f(r), b \oplus h(r)$. This is the **commit phase**. Let the commitment be denoted by C.
The verifier will check $C = (f(r), b \oplus h(r))$, will accept if its true or reject otherwise. This is the **open phase**.

Testing security: As f is OWP, first part of the commitment fixes exactly one value of r. Thereby, fixing $h(r)$ would fix b as $h(r)$ is deterministic.

5 Revisit Zero Knowledge Proof for Graph 3-Coloring

Protocol (P, V): Prover P, Verifier V, common input $G = (E, V)$ where $|V| = n$ and $color_1, color_2..color_n \in \{1, 2, 3\}$ are the colors for the nodes of the graph, repeats the following procedure $n|E|$ times using fresh randomness each time:

- $P \to V$: Prover P chooses a random permutation π over {1,2,3}. For every $i \in [n]$, it computes $C_i = Com(\widetilde{color_i})$ where $\widetilde{color_i} = \pi(color_i)$. It sends $(C_1, C_2 \dots C_n)$ to the verifier V
- $V \to P$: V chooses a random edge $(i, j) \in E$ and sends it to P
- $P \to V$: Prover P opens C_i and C_j to reveal $(\widetilde{color_i}, \widetilde{color_j}$
- V : If the openings of C_i, C_j are valid and $\widetilde{color_i} \neq \widetilde{color_j}$, then V accepts the proof and rejects, otherwise.

5.1 Proof of Soundness

If it is not 3-colorable, then it will not be colored correctly. Meaning, there will be at least one edge that has the same color on both the endpoints. From the binding property, we know that $C_1, \dots, C_n$ have unique openings $\widetilde{color_1}, \dots, \widetilde{color_n}$. The verifier V chooses $i = i^*$ and $j = j^*$ with probability $\frac{1}{|E|}$ to catch P where $(i^*, j^*) \in E$ such that $\widetilde{color_i^*} = \widetilde{color_j^*}$. This means, for $n|E|$ repetitions, P can successfully cheat with probability at most

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

5.2 Proving Zero Knowledge

Intuitively, zero knowledge can be proved by using the hiding property of Com as it guarantees that everything remains hidden from V except the two colors of the edge he picks. Let us see how that can be put in a more formal way.

5.2.1 Construction of Simulator S

Algorithm 2 Simulator $S(x = G, z)$:

1: Choose a random edge $(i^{'}, j^{'}) \overset{\$}{\leftarrow} E$ and pick random colors $color^{'}_{i^{'}}, color^{'}_{j^{'}} \overset{\$}{\leftarrow}$ {1,2,3} s.t. $color^{'}_{i^{'}} \neq color^{'}_{j^{'}}$ and set $color^{'}_{k} = 1$ for every other $k \in [n] \backslash \{i^{'}, j^{'}\}$
2: Compute $C_l = Com(color^{'}_{l})$ for every $l \in [n]$
3: Emulate execution of $V^*(x, z)$ by giving $(C_1 \dots C_n)$ as input. Let (i, j) be the output.
4: If $(i, j) = (i^{'}, j^{'})$
    feed the openings of C_i, C_j to V^* and output its view
   else
    restart with Step 1 at most $n|E|$ times
5: If simulation does not succeed after $n|E|$ attempts, then output **fail**

5.2.2 Correctness of Above Simulation

We will prove the correctness of the simulation using hybrid arguments where we will change the witness one at a time till the point when no witness is required by the simulator machine.

- H_0 : Real execution with the prover machine P
- H_1 : Hybrid Simulator $S^{'}$ that chooses $(i^{'}, j^{'}) \overset{\$}{\leftarrow} E$ at random along with the witnesses $color_1 \dots color_n$ and outputs **fail** if $(i^{'}, j^{'}) \neq (i, j)$
  If $S^{'}$ does not output **fail**, then H_0 and H_1 are proved to be identical. H_0 and H_1 seem to be statistically indistinguishable as (i, j) and $(i^{'}, j^{'})$ are independently chosen and $S^{'}$ will fail with least probability of
  $$\left(1 - \frac{1}{|E|}\right)^{n|E|} \approx e^{-n}$$
  which follows that $H_0 \approx H_1$
- H_2 : Simulator machine S which does not require witnesses at all.
  For all $k \in [n] \backslash \{i^{'}, j^{'}\}$, C_k is a commitment to $\pi(color_k)$ in H_1 and a commitment to 1 in H_2. Multi-value hiding property shows that $H_1 \approx H_2$.