CSE 594 : Modern Cryptography 04/06/2017

Lecture 18: Zero Knowledge Proofs

Instructor: Omkant Pandey *Scribe: Malini Mahalakshmi Venkatachari, Ravikumar Rajendran*

1 Proof

Proof is an argument (or sufficient evidence) that can convince a reader of the truth of some statement. In Mathematics, a proof can be stated as a deductive argument for a statement, by reducing the validity of the statement to a set of axioms or assumptions.
Following are the desirable features in a proof.

1. The verifier should accept the proof if the statement is true
2. The verifier should reject any proof if the statement is false
3. Proof must be finite (or succinct) and efficiently verifiable

For an example a proof that there are infinitely many primes should not simply be a list of all the primes. Not only would it take forever to generate that proof, it would also take forever to verify it. It should be finite and we should able to verify it efficiently. Verifier must be polynomial time in the length of the statement. A proof can be non-interactive as well as it can be a conversation.

2 Interactive Protocol

Interactive Turing Machine (ITM) is a Turing machine with two additional tapes: a read-only communication tape for receiving messages, a write-only communication tape for sending messages. An interactive protocol (M1, M2) is a pair of ITMs that share communication tapes $\ni$ the send-tape of the first ITM is the receive-tape of the second, and vice-versa.Protocol proceeds in rounds. In each round, only one ITM is active, the other is idle. Protocol ends when both ITMs halt. Following denotes various parts of the protocol :

1. M1(x1, z1) $\longleftrightarrow$ $M2(x2, z2)$: A randomized protocol execution with x_i as input and z_i as auxillary input of M_i can be depicted as below:
2. $Out_{M_i}(e)$: Output of M_i in an execution e
3. $View_{M_i}(e)$: View of M_i in an execution e consists of its input, random tape, auxiliary input and all the protocol messages it sees.

3 Interactive Proofs

Definition 1 *Interative Proofs : A pair of ITMs (P, V) is an interactive proof system for a language L if V is a PPT machine and the following properties hold:*

1. ***Completeness :*** *For every $x \ni L$, $Pr[Out_V[P(x) \longleftrightarrow V(x)] = 1] = 1$*

2. ***Soundness :*** *There exists a negligible function v(.) $\ni \forall x \notin L$ and for all adversarial provers P^* $Pr[Out_V[P^*(x) \longleftrightarrow V(x)] = 1] \leq v(|x|)$*

4 Interactive Proofs - Motivation

Let L be a language in NP and let R be the associated relation. For any $x \in L$, there exists a small (polynomial-size) witness w.By checking that R(x, w) = 1, we can verify that x ∈ L. Therefore, w is a non-interactive proof for x. For example let us consider Graph Isomorphism. Two graphs G0 and G1 are isomorphic if there exists a permutationπ that maps the vertices of G0 onto the vertices of G1. If that is the case we need to look at why interactive proofs are actually required. We can state the below as two main reasons for interaction :

- Proving statements in languages not known to be in NP

  Single prover [IP=PSPACE]

  Multiple provers [MIP = NEXP]

- Achieving privacy guarantee for prover. In Zero knowledge proofs, prover learns nothing from the proof beyond the validity of the statement.

  Let us look at example of interactive proofs for graph isomorphism.

5 Graph Isomorphism

Notation for Graphs:

Below are the various notations for graphs

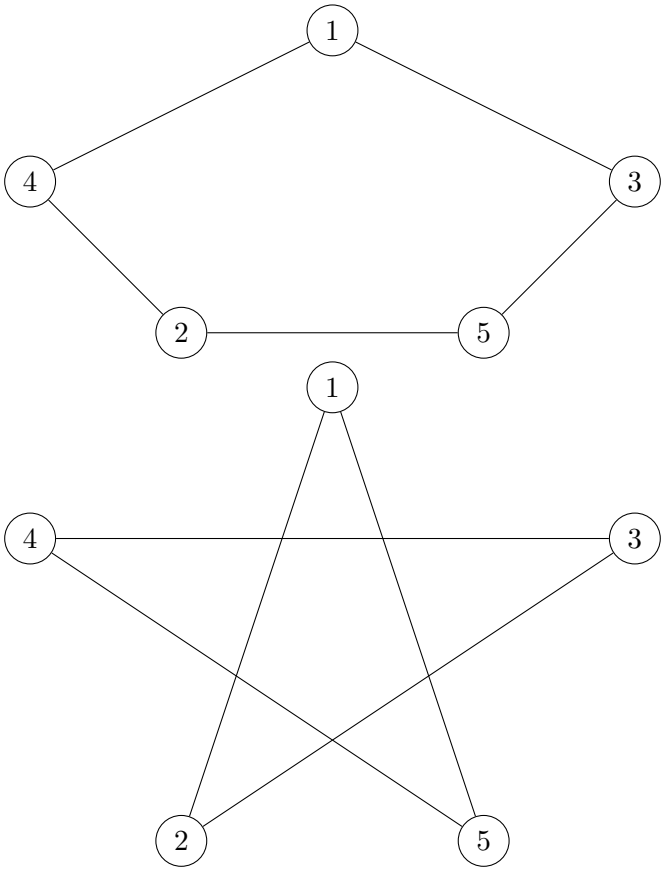
- Graph G = (V, E) where V is set of vertices and E is set of edges
- $|V| = n, |E| = m$
- $\pi_n \longrightarrow$ set of all permutations π over n vertices
- To graphs $G_0 = V_0, E_0$ and $G_1 = V_1, E_1$ are said to be isomorphic if there exist a permutation $\pi \ni$

  $V_1 = pi(v)|v \in V_0$

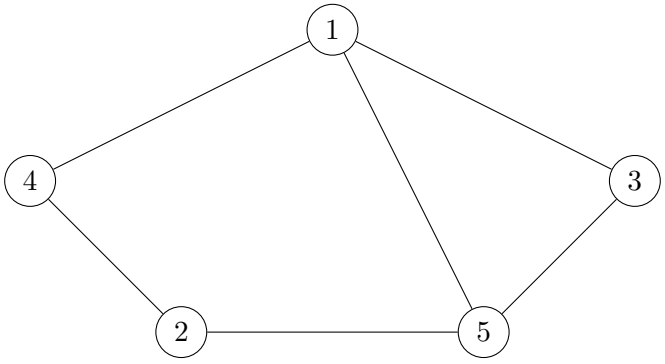
  $E_1 = (pi(v_1), pi(v_2))|(v_1, v_2 \in E_0)$

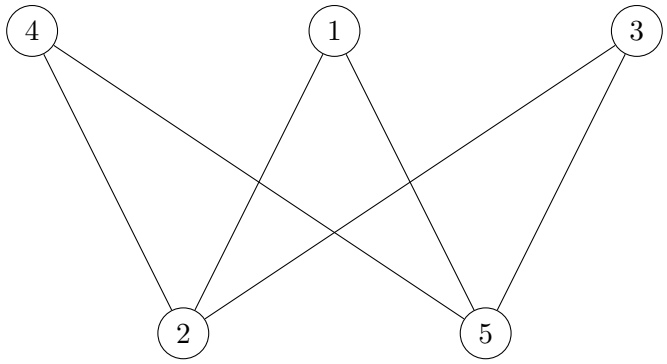
  Alternatively $G_1 = \pi(G_0)$

- Further it should be noted that Graph Isomorphism belongs to NP



- Graph Non-Isomorphism: G_0 and G_1 are non-isomorphic if there exists no permutation $\pi \in \pi_n \ni G_1 = \pi(G_0)$





- Graph Non-Isomorphism is in co-NP and not known to be in NP.

5.1 Proof for Graph Non-Isomorphism

Suppose P wants to prove that G_0 and G_1 are not isomorphic. One way to prove this is to write down all possible permutations π over n vertices and show that for every π, $G_1 \neq \pi(G_0)$. But this is not efficiently verifiable. This calls for interactive proof that is efficiently verifiable.

5.2 Interactive Proof for Graph Non-Isomorphism:

Both the verifier and prover gets a common input that is G_0, G_1.

Protocol(P,V) : Repeat the following procedure n times using fresh randomness

1. V $\longrightarrow$ P : V chooses a random bit $b \in 0, 1$ and a random permutation $\pi \in \pi_n$. It computes $H = \pi(G_b)$ and sends H to P
2. P $\longrightarrow$ V : P computes $b^{'} \ni$ H and $G_{b'}$ are isomorphic and sends $b^{'}$ to V.
3. V $(x, b, b^{'})$: V outputs 1 if $b^{'} = b$ and 0 otherwise

Here (P,V) is an Interactive proof.It can be shown to hold the following two properties:

- **Completeness:** If G_0 and G_1 are not isomorphic, then an unbounded power can always find $b^{'} \ni b^{'} = b$
- **Soundness:** : If G_0 and G_1 are isomorphic, then H is isomorphic to both G_0 and G_1. Therefore, in one iteration, any (unbounded) prover can correctly guess b with probability at most $1/2$. Since each iteration is independent, prover can succeed in all iterations with probability at most 2^{-n}

6 Interactive Proofs with Efficient Provers

1. In the case of Interactive proof system of Graph Non-Isomorphism, Prover is inefficient. This must be the case because the problem of Graph Non-Isomorphism would come into the class of **NP** otherwise.

2. But, when we want Interactive Proofs with efficient Provers, we should restrict our attention to languages in NP. Prover must employ an efficient strategy when it is given a witness w for a statement x that it attempts to prove.

Definition 2 *An interactive proof system (P, V) for a language L with witness relation R is said to have an efficient prover if P is PPT and the completeness condition holds for every $w \in R(x)$.*

Remark 1 *Despite the fact that the honest Prover P is efficient, the soundness requirement still must hold good against all adversarial Provers.*

7 Interactive Proof for Graph Isomorphism

1. Recall that to prove Graphs G_0 and G_1 are isomorphic, Prover P can simply send the permutation π such that $G_1 = \pi(G_0)$. Additionally, if P is given the permutation π then P is efficient as well.

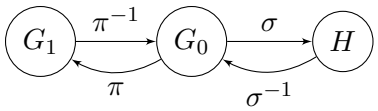
2. Despite efficient P, Verifier V learns of the witness π in the above case. Further V can use it to prove the isomorphism between G_0 and G_1 to someone else.

3. This immediately gives rise to the question of whether we can construct an interactive proof system that hides the witness π from V.

4. Further, it can be extended as whether we can construct an interactive proof system that reveals only the validity of the statement to V and nothing else. **Goldwasser, Micali, Rackoff** showed that it can indeed be done!

Common Input: $x = (G_0, G_1)$
P's witness: Permutation π such that $G1 = \pi(G_0)$
Protocol (P, V)**:** The following procedure is repeated n times using fresh randomness.

$P \to V$: Prover chooses a random permuation $\sigma \in \Pi_n$, computes $H = \sigma(G_0)$ sends H to V.
$V \to P$: V chooses a random bit $b \in \{0, 1\}$ and sends it to P.
$P \to V$: If $b = 0$, P sends $\phi = \sigma$. Otherwise, it sends $\phi = \sigma.\pi^{-1}$
$V(x, b, \phi)$: V outputs 1 iff $H = \phi(G_b)$.



7.1 (P, V) is an Interactive Proof

- **Completeness:** If G_0 and G_1 are isomorphic, V would always accept since $\sigma(G_0) = H$ (when $b = 0$) and $\sigma(\pi^{-1}(G_1)) = \sigma(G_0) = H$ (when $b = 1$).

- **Soundness:** If G_0 and G_1 are not isomorphic, then H is isomorphic to either G_0 or G_1, but not to both. Since the bit b is chosen at random after H is fixed, with probability $\frac{1}{2}$, H is not isomorphic to G_b. Thus, an adversarial prover can succeed with probability at most $\frac{1}{2}$. Since each iteration is independent, prover can succeed in all iterations with probability at most 2^n.

8 Towards Zero Knowledge

- The above Graph Isomorphism Interactive Proof system also has the property that the verifier V does not gain any knowledge from its interaction with P beyond the fact that G_0 and G_1 are isomorphic.
- In particular, P's witness w remains private from V.

8.1 Questions on formalizing the notion

1. How to formalize the notion of "V does not gain any knowledge from its interaction with P"?
2. What in fact is knowledge?

8.2 Rules for formalizing Zero Knowledge

1. Randomness is for free
2. Polynomial time computation is for free

The above rules imply that there is no knowledge gained by learning the result of a random process or the result of a polynomial time computation.

9 When is knowledge conveyed?

Scenario 1: Someone tells you that she will sell you a 100-bit random string for \$1000

Scenario 2: Someone tells you that she will sell you the product of two prime numbers of your choice for \$1000

Scenario 3: Someone tells you that she will sell you the output of an exponential time computation (eg. checking isomorphism between 2 graphs) for \$1000

In the 1^{st} and 2^{nd} cases, we can do the computation on our own for free - Since both generation of 100-bit random string and product of two given prime numbers are efficiently computable in polynomial time. On the other hand, exponential time computation is beyond the limits of our capacity as a PPT. Thus, serious consideration can only be given for the 3^{rd} one.

10 Zero Knowledge

10.1 Intuition

- V can generate a protocol transcript on its own, without talking to P. If this transcript is indistinguishable from a real execution, then clearly V does not learn anything by talking to P.
- The above intuition is formalized via notion of *Simulator*, as in definition of semantic security for encryption.

10.2 Definition I (Honest Verifier Zero Knowledge)

An interactive proof (P, V) for a language L with witness relation R is said to be *honest verifier zero knowledge* if there exists a PPT simulator S such that for every non-uniform PPT distinguisher D, there exists a negligible function $\nu(\cdot)$ such that for every $x \in L$, $w \in R(x)$, $z \in \{0,1\}^*$, D distinguishes between the following distributions with probability atmost $\nu(n)$:

- $\Big\{View_V[P(x, w) \iff V(x, z)]\Big\}$
- $\Big\{S(1^n, x, z)\Big\}$

10.3 Remarks on Definition I

- The definition clearly captures that whatever V could have generated all of what it "saw" in the interactive proof on its own by running the simulator S.
- The auxiliary input z to the verifier V captures any a priori information V might have about x. Thus the promise of V does not learn anything new is still intact.
- **Problem:** However, the above promise holds only if the verifier V follows the protocol. The same is not true incase if V is malicious and deviates from the honest strategy.
- **Want:** A simulator S for every possibly malicious efficient verifier strategy V^*.
- For now, we can relax the simulator to be an expected PPT thus it can be thought of as a machine whose expected running time is polynomial.

10.4 Definition II (Zero Knowledge)

An interactive proof (P, V) for a language L with witness relation R is said to be *zero knowledge* if for every non-uniform PPT adversary V^*, there exists an expected PPT simulator S such that for every non-uniform PPT distinguisher D, there exists a negligible function $\nu()$ such that for every $x \in L$, $w \in R(x)$, $z \in \{0,1\}^*$, D distinguishes between the following distributions with probability atmost $\nu(n)$:

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

10.5 Remarks on Definition II

- If the distributions are statistically close, then we call it *statistical zero knowledge*
- On the other hand, if the disributions are identical, then we call it *perfect zero knowledge*