Introduction to Modern Cryptography
Lecture 10

January 3, 2017

Instructor: Benny Chor
Teaching Assistant: Orit Moskovich

School of Computer Science
Tel-Aviv University

Fall Semester, 2016–17
Tuesday 12:00–15:00
Venue: Meron Hall, Trubowicz 102 (faculty of Law)

Course site: http://tau-crypto-f16.wikidot.com/

Lecture 10: Plan

- Interactive proof systems.
- Zero knowledge proof systems

Some of this class will be presented using the decade old technique of whiteboard and markers.

And Now to Something Completely Different: Interactive Proof Systems

תֶּחֱזַקְנָה יְדֵי כָל-אַחֵינוּ הַמְחוֹנְנִים
עַפְרוֹת אַרְצֵנוּ בַּאֲשֶׁר הֵם שָׁם;
אַל יִפֹּל רוּחֲכֶם – עַלִּיזִים, מִתְרוֹנְנִים
בֹּאוּ שְׁכֶם אֶחָד לְעֶזְרַת הָעָם!

– חיים נחמן ביאליק, **ברכת עם**

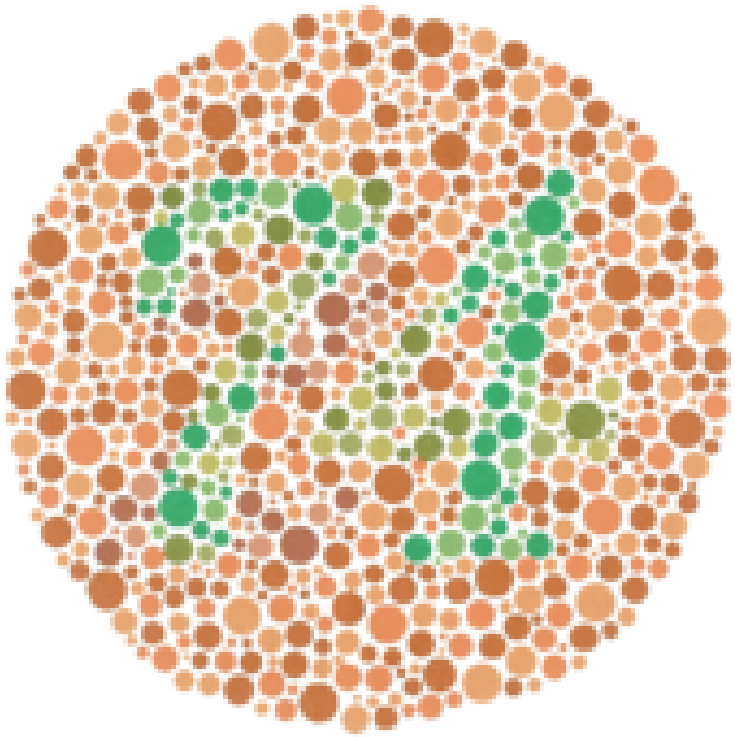
Interactive and Zero Knowledge Proofs: Plan

- Traditional proof systems and their relations to the class NP.
- Interactive proof systems.
- An interactive proof for graph non isomorphism (a problem in coNP that is not known to be in NP).
- Zero knowledge interactive proof systems.
- A zero knowledge proof for graph isomorphism.
- A zero knowledge proof for graph 3 colorability (an NP complete problem), using commitment schemes[1].

[1] which can be built using pseudo random generators, or, equivalently, one way functions.

Interactive Proofs: The Two Apples Story

Benny wants to convince the color blind[2] Adam (staaaaaam) that two apples with identical physical characteristic (shape, mass, etc.) have different colors.



[2] Deuteranopia (1% of males), Tritanopia (less than 1% of males), Protanomaly (1% of males), Deuteranomaly (most common – 6% of males).

Interactive Proofs: The Two Apples Story

Benny wants to convince the color blind Adam that two apples with identical physical characteristic (shape, mass, etc.) have different colors.

We will boldly attempt such protocol, live, in class.

Proofs

- In mathematics and in life, we often want to convince or prove things to others.
- Typically, if I know that X is true, and I want to convince you of that, I try to present all the facts I know and the inferences from that fact that imply that X is true.
- Example: I know that 26781 is not a prime since it is $113 \cdot 237$, to prove to you that fact, I will present these factors, and demonstrate that indeed $113 \cdot 237 = 26781$.

Interactive Proofs

We will use a ppt presentation by Prof. Muli Safra (much fancier than those you are used to in our humble course) about interactive proofs.

Zero-Knowledge Proofs – Goldwasser, Micali and Rackoff

- Typically, a proof yields some knowledge, beyond the fact that the statement is true.
- In the previous example, we learned that 26781 is not a prime, and, in addition, we learned its factorization, $113 \cdot 237$.
- Zero knowledge proof tries to avoid it.

Zero-Knowledge Proofs – Goldwasser, Micali and Rackoff

- Typically, a proof yields some knowledge, beyond the fact that the statement is true.
- In the previous example, we learned that 26781 is not a prime, and, in addition, we learned its factorization, $113 \cdot 237$.
- Zero knowledge proof tries to avoid it.

Intuitively:

Zero-Knowledge Proofs (GMR 85')

Alice will prove to Bob that a statement X is true, Bob will be convinced that X is true, but will not learn anything as a result of this process.

Zero-Knowledge Proofs – Goldwasser, Micali and Rackoff

- Typically, a proof yields some knowledge, beyond the fact that the statement is true.
- In the previous example, we learned that 26781 is not a prime, and, in addition, we learned its factorization, $113 \cdot 237$.
- Zero knowledge proof tries to avoid it.

Intuitively:

Zero-Knowledge Proofs (GMR 85')

Alice will prove to Bob that a statement X is true, Bob will be convinced that X is true, but will not learn anything as a result of this process.

One of the most beautiful and influential concepts in CS.
Lead to many applications (*e.g.* practical digital signatures, and hardness of approximation).

Zero Knowledge Proofs

And now we turn to a a pdf presentation by Prof. Maurice (also much fancier than those you are used to in our humble course), demonstrating a zero knowledge proof for graph 3 colorability.

Application 1: Identification

- We want to control access to the EE department.
- Solution: Give people with entry authorization a smart card with a PIN, and put a box outside the building that verifies the PIN.
- Problem: Box is outside! Someone may attack it and discover the PIN.
  - by reading the memory.
  - by installing a fake box that records the user's PIN.
- Better if the box contains no secret information at all.

Application 1: Identification

- We want to control access to the EE department.
- Solution: Give people with entry authorization a smart card with a PIN, and put a box outside the building that verifies the PIN.
- Problem: Box is outside! Someone may attack it and discover the PIN.
  - by reading the memory.
  - by installing a fake box that records the user's PIN.
- Better if the box contains no secret information at all.

Solution: Let the box store $f(PIN)$ where f is one-way function. The user proves in ZK to the Box that he knows PIN.

Application 2: Protocol Design

- Alice & Bob, who don't trust each other, run some crypto protocol.
- "Security" holds if Alice and Bob follow the instructions.
  - e.g., Alice should choose an RSA modulus $n = pq$.
- But what if Alice does not follow the protocol ?
  - e.g., chooses $n = pqr$.
- Security may be lost !
- Bad Solution: Alice sends her inputs and let Bob verify that all is well
  - e.g., reveals n, p, q.
- This is bad for Alice: she does not trust Bob, and the factorization of $n = pq$ should remain private !

Application 2: Protocol Design

- Alice & Bob, who don't trust each other, run some crypto protocol.
- "Security" holds if Alice and Bob follow the instructions.
  - e.g., Alice should choose an RSA modulus $n = pq$.
- But what if Alice does not follow the protocol ?
  - e.g., chooses $n = pqr$.
- Security may be lost !
- Bad Solution: Alice sends her inputs and let Bob verify that all is well
  - e.g., reveals n, p, q.
- This is bad for Alice: she does not trust Bob, and the factorization of $n = pq$ should remain private !

Solution: Alice proves to Bob that she followed the instructions of the protocol correctly using Zero-Knowledge Proofs.

Zero Knowledge Proofs

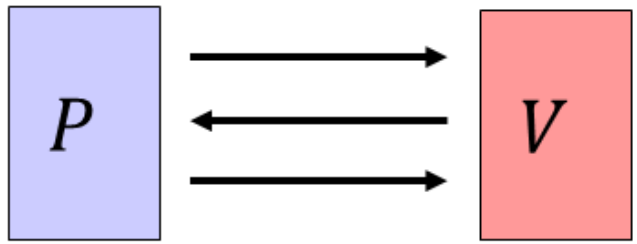
Can I convince you that some statement $\mathcal{S}$ holds, without giving you any hint about its proof?

- Not by authority or intimidation!
- By some proof system, possibly randomized, where
- If $\mathcal{S}$ is false, the probability that I convince you is smaller than ε (a small parameter).
- If $\mathcal{S}$ is true, the probability that I convince you is larger than $1 - \varepsilon$.
- The text of the conversations, assuming $\mathcal{S}$ is true, gives you nothing more – you could have generated very similar text on your own.

- We just saw a zero knowledge proof for graph 3 colorability.
- We will present (on board) a zero knowledge proof for graph isomorphism.
- We will present (on board) a zero knowledge proof of knowledge for an instance of the discrete logarithm problem.

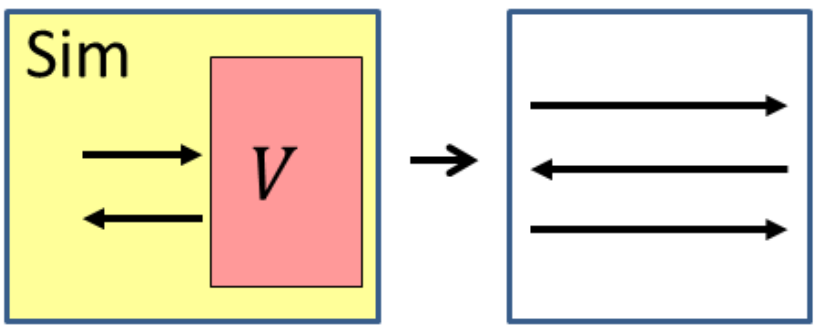
Zero Knowledge: Ideal Scenario vs. Real Scenario (images by the same good old anonymous referee)

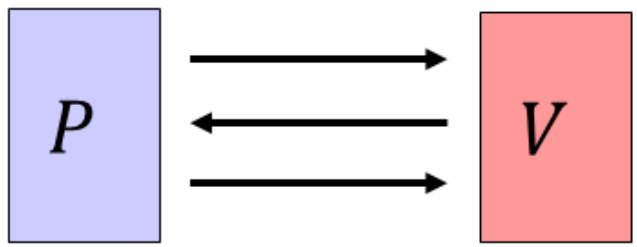
The actual scenario: What does the verifier observe in an interactive proof system?



Zero Knowledge: Ideal Scenario vs. Actual Scenario, cont. (images by the same anonymous referee)

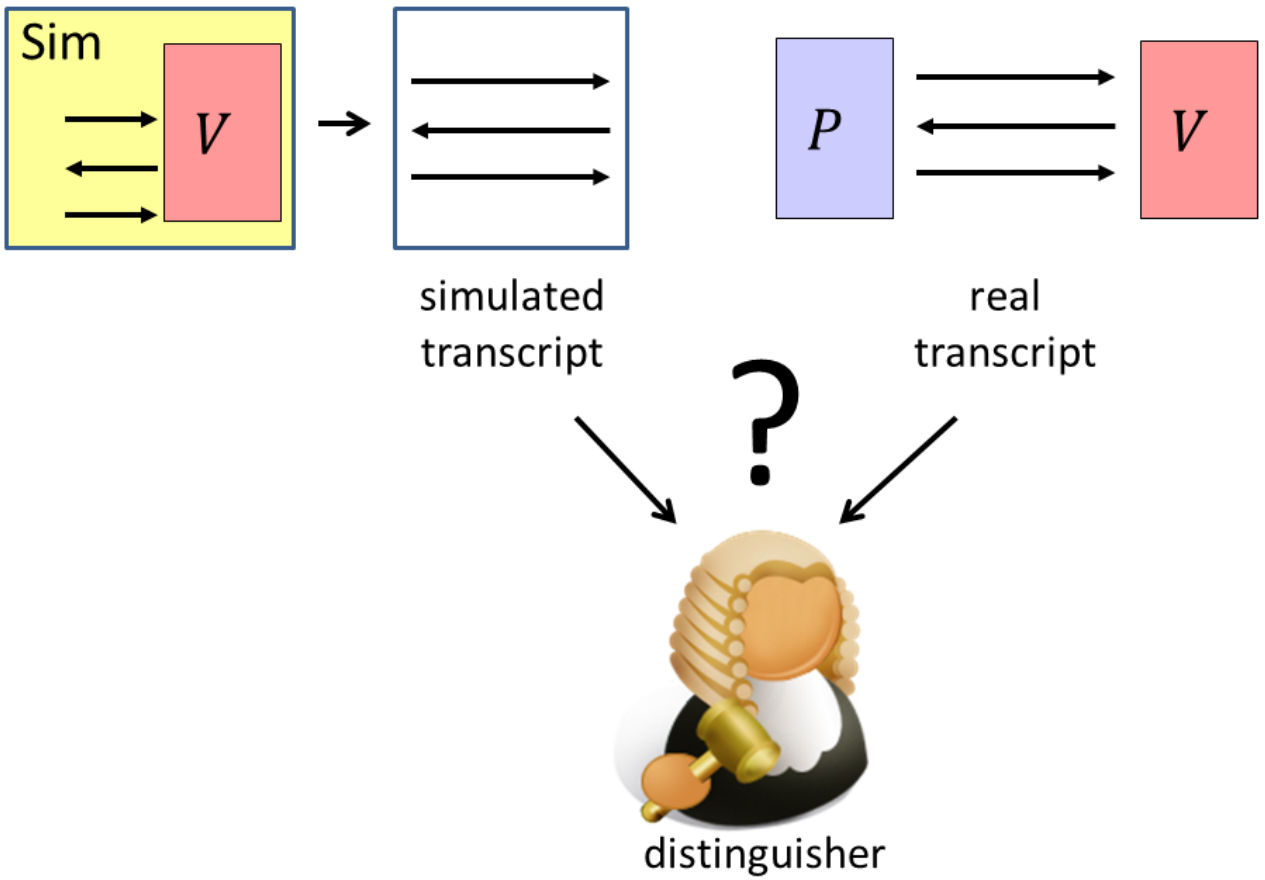
The simulated scenario: The simulator (which can query the verifier, run it, make it work, probe its cortex, etc.) produces a simulated transcript of the communication with the prover.





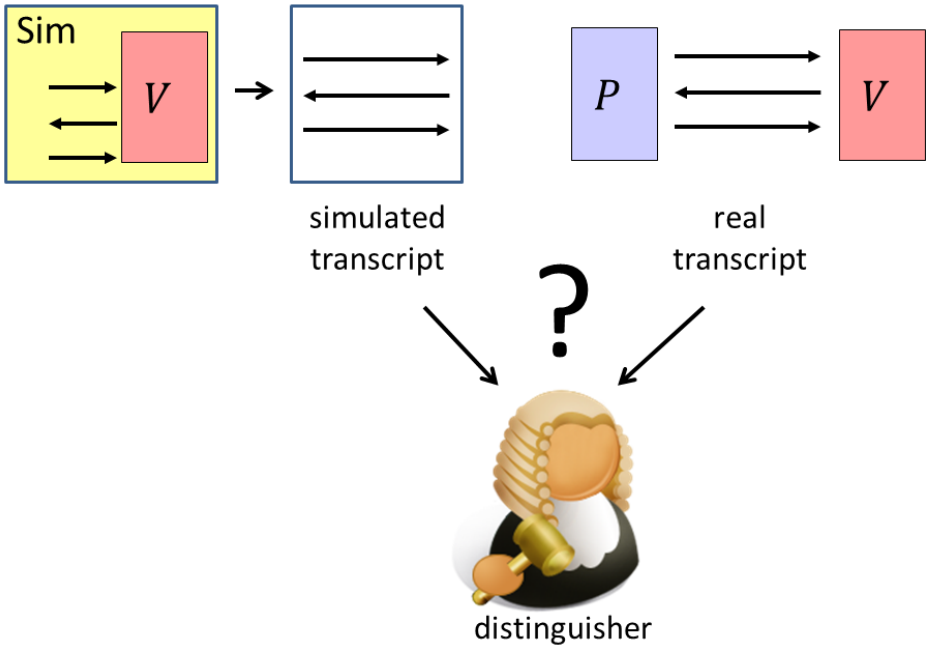
Zero Knowledge: Ideal Scenario vs. Actual Scenario, cont.

Actual communication vs. simulated communication:
A distinguisher cannot tell the two ensembles of communication apart.



Distinguisher is either computationally bound, or (stronger) unbounded (statistically indistinguishability).

Zero Knowledge: Ideal Scenario vs. Actual Scenario, cont.



Distinguisher is either computationally bound, or (stronger) unbounded (statistically indistinguishability).

ZK proof for 3 colorability: Computational indistinguishability (due to use of encryption/envelopes).

The other ZK proofs we saw yield perfect indistinguishability (without compromising low prob. of cheating).

Honest-Verifier Zero-Knowledge Proofs

The view $\text{view}_V(x)$ of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Honest-Verifier Zero-Knowledge Proofs

The view $\text{view}_V(x)$ of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Definition (Honest Verifier Perfect Zero-Knowledge Proofs)

An HVZK system for a language L is a proof system (P, V) that has an efficient simulator S that runs in expected polynomial time such that for every $x \in L$

$$S(x) \equiv \text{view}_V(x).$$

Honest-Verifier Zero-Knowledge Proofs

The view $\text{view}_V(x)$ of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Definition (Honest Verifier Perfect Zero-Knowledge Proofs)

An HVZK system for a language L is a proof system (P, V) that has an efficient simulator S that runs in expected polynomial time such that for every $x \in L$

$$S(x) \equiv \text{view}_V(x).$$

Note: Zero-knowledge is required to hold only for $x \in L$!

Honest-Verifier Zero-Knowledge Proofs

The view $\text{view}_V(x)$ of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Definition (Honest Verifier Perfect Zero-Knowledge Proofs)

An HVZK system for a language L is a proof system (P, V) that has an efficient simulator S that runs in expected polynomial time such that for every $x \in L$

$$S(x) \equiv \text{view}_V(x).$$

Note: Zero-knowledge is required to hold only for $x \in L$!

Later today: define ZK against a cheating verifier

Zero-Knowledge Proofs – Definition

The view, $\text{view}_V(x)$, of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Zero-Knowledge Proofs – Definition

The view, $\text{view}_V(x)$, of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Definition (Honest Verifier Perfect Zero-Knowledge Proofs)

An HVZK system for a language L is a proof system (P, V) that has an efficient simulator S that runs in expected polynomial time such that for every $x \in L$

$$S(x) \equiv \text{view}_V(x).$$

Zero-Knowledge Proofs – Definition

The view, $\text{view}_V(x)$, of the verifier V in a proof system is a random variable that consists of the public input x, the internal randomness of V, and the incoming messages.

Definition (Honest Verifier Perfect Zero-Knowledge Proofs)

An HVZK system for a language L is a proof system (P, V) that has an efficient simulator S that runs in expected polynomial time such that for every $x \in L$

$$S(x) \equiv \text{view}_V(x).$$

- Simulator is allowed to run in expected polynomial time
- $\equiv$ means that the two r.v.'s are identically distributed
- Computational variant: $S(x)$ and $\text{view}_V(x)$ are (t, ε) indistinguishable.

Note on Honest Verifier vs. Malicious Verifier

Often the honest verifier enables a simpler simulation strategy.

In schemes involving commitments (like the envelopes in graph 3 colorability), can use hiding to argue simulator succeeds with non negligible probability even for malicious verifiers that deviate from protocol (since they are poly time machines).

We hope to discuss these issues later today.

Zero Knowledge Proof for Graph Isomorphism (sketch)

Setting: Two graphs G_1, G_2 such that $G_2 = \varphi(G_1)$.

The prover knows the homomorphism φ. Wishes to convince the verifier that the two are indeed isomorphic without revealing φ.

Repeat 100 times:

Prover: Choose at random a permutation ψ of the nodes of G_1. Generate the induced graph, $H = \psi(G_1)$. Send $H = \psi(G_1)$ to verifier.

Verifier: Flip a coin. If heads, send 1 to prover;
if tails, send 2 to prover.

Prover: Send accordingly either a mapping (permutation) ψ^{-1}, mappling $H \mapsto G_1$, or $\varphi \circ \psi^{-1}$, mapping $H \mapsto G_2$ to verifier.

Verifier: Checks and accepts/rejects.

Note that in both cases, verifier sees a random permutation.

For the honest verifier, simulator can generate a transcript which is distributed exactly as the interaction above (without knowledge of the homomorphism).

Zero Knowledge Proof of Knowledge of a Discrete Log (sketch)

Setting: Public prime p, multiplicative generator $g \in Z_p^*$. Bob publishes $g^x \pmod p$ and tells the post office to hand deliver packages addressed to Bob to anyone who can prove s/he knows x.

Bob is the prover. The post office is the verifier.

Repeat 100 times:

Prover: Choose at random $r, 0 \leq r \leq p-2$, send $g^r \pmod p$ to verifier.
Verifier: Flips a coin, b and sends it to prover. Prover: If heads, send r to prover; if tails, send $r + x \pmod{p-1}$ to verifier.
Verifier: Accepts in this round if exponent fits the corresponding power.

For the honest verifier, simulator can generate a transcript which is distributed exactly as the interaction above (without knowledge of the discrete logarithm, x).

Non Zero Knowledge Interactive Proofs

Consider the following protocols:

- Public input: $n = p \cdot q$. Bob proves n is composite by revealing p.
- Public input: $p, g^x \pmod{p}$. Bob proves he knows x by revealing it.

Convince yourself these two protocols are not zero knowledge.

Interactive Proofs: The Two Apples Story, cont.

Benny wants to convince the color blind Adam (staaaaaam) that two apples with identical physical characteristic (shape, mass, etc.) have different colors.

Easy challenge: Modify the protocol so that it is also zero knowledge, namely poor, color blind Adam only learns the two apples have different colors, but not which color each apple is.

Zero-Knowledge Proofs with Cheating Verifier

Definition (Zero-Knowledge Proofs)

A ZK system for a language L is a proof system (P, V) that for every efficient verifier V^* there exists an efficient simulator $S = S_{V^*}$ such that for every $x \in L$

$$S^*(x) \equiv \text{view}_{V^*}(x).$$

Zero-Knowledge Proofs with Cheating Verifier

Definition (Zero-Knowledge Proofs)

A ZK system for a language L is a proof system (P, V) that for every efficient verifier V^* there exists an efficient simulator $S = S_{V^*}$ such that for every $x \in L$

$$S^*(x) \equiv \text{view}_{V^*}(x).$$

- Simulator is allowed to run in expected polynomial time
- $\equiv$ means that the two r.v.'s are identically distributed
- Computational variant: $S^*(x)$ and $\text{view}_{V^*}(x)$ are (t, ε) indistinguishable.