The Evolution of Proofs in Computer Science:

Zero-Knowledge Proofs

6.857
Lecture 13
Classical Proofs
Conjecture: \( \forall \) succinct classical proof for correctness of any computation \( M(x) = 1 \) within \( T \) steps
Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff85]

Proofs that reveal no information beyond the validity of the statement.
Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff85]

Impossible!

This is information!
Interactive Proofs
[Goldwasser-Micali-Rackoff85]

Completeness: \( \forall x \in L \ \Pr[(P, V)(x) = 1] \geq 2/3 \)

Soundness: \( \forall x \notin L, \forall P^* \ \Pr[(P^*, V)(x) = 1] \leq 1/3 \)

Note: By repetition, we can get completeness \( 1 - 2^{-k} \), and soundness \( 2^{-k} \)
Interactive Proofs

[Goldwasser-Micali-Rackoff85]:

For ZK the prover needs to be randomized

\[ P \rightarrow V \]

\[ V \rightarrow P \]

\[ P \rightarrow V \]

\[ V \rightarrow P \]

[Goldreich-Micali-Wigderson87]: Every statement that has a classical proof has zero-knowledge (ZK) interactive proof, assuming one-way functions exist.
Defining Zero-Knowledge

Formally: There exists a \( PPT \) algorithm \( S \) (called a simulator), such that for every \( x \in L \):

\[
S(x) \approx (P, V)(x)
\]

\( P \) \( \xrightarrow{x \in L} \) \( V \)

This transcript reveals \textit{no information}

Denotes the transcript
ZK Proofs for NP

For the NP-complete language of all 3-colorable graphs

\[ G = (V, E) \]

Randomly permute the coloring, to obtain valid coloring \((c_1, ..., c_n)\)

Choose a random edge \((i, j) \in E\)

Open safes \(i, j\)

**Soundness:** Only \(1 - \frac{1}{|E|}\) but can be amplified via repetition.
For the $NP$-complete language of all 3-colorable graphs

$$G = (V, E)$$

**$S(V, E)$:**

1. Choose a random $(i, j) \in E$
2. Choose random distinct colors $c_i, c_j$
3. The simulated transcript is: $i, j$ have values $c_i, c_j$

Open safes $i, j$
Implementing Digital Safes: Commitment Scheme

A commitment scheme is a randomized algorithm $Com$ s.t.:

- **Hiding:** $\forall m, m' \; Com(m; r) \approx Com(m'; r')$.

- **Binding:** $\not\exists (m, r), (m', r')$ s.t. $m \neq m'$ and $Com(m; r) = Com(m'; r')$.
Using Commitments to Construct ZK Proofs

For the \( NP \)-complete language of all 3-colorable graphs

\[
G = (V, E)
\]

**Randomly permute the coloring, to obtain valid coloring \((c_1, \ldots, c_n)\)**

**Choose a random edge \((i, j) \in E\)**

**Reveal \(c_i, c_j\), with corresponding randomness**
Constructing a Commitment Scheme

Construction 1:
Let $f : \{0,1\}^* \rightarrow \{0,1\}^*$ be an injective OWF.

$$\text{Com}(b; (r, s)) = (f(r), s, (\oplus r_i s_i) \oplus b)$$

**Binding:** Follows from the fact that $f$ is injective

**Hiding:** Relies on the fact that if $f$ is one-way then:

$$(f(r), s, \oplus r_i s_i) \approx (f(r), s), U)$$

Known as a **hard-core predicate**

[Goldreich-Levin89]
Constructing a Commitment Scheme

Construction 2:
Let $G$ be a group of prime order $p$, let $g \in G$ be any generator, and $h$ be a random group element.

$$Com_{g,h}(m,r) = g^m h^r$$

Hiding: Information theoretically!

Binding: Follows from the Discrete Log assumption.
If $\exists PPT$ alg $A$ s.t.

$A(g,h) = (m_1, m_2, r_1, r_2)$ where $g^{m_1} h^{r_1} = g^{m_2} h^{r_2}$ then

$m_1 + sr_1 = m_2 + sr_2 \mod p$,

which implies that $s = \frac{m_1 - m_2}{r_2 - r_1} \mod p$
Constructing Zero Knowledge Proofs

This is **perfect ZK!**
But only **computationally sound**

Randomly permute the coloring, to obtain valid coloring $(c\#_1, \ldots, c\#_n)$. 

Choose a random edge $(i, j) \in E$. 

Reveal $c_i, c_j$, with corresponding randomness.

Perfectly hiding

All powerful prover can break binding
Interactive Computationally Sound Proofs
(a.k.a. Arguments)
[Brassard-Chaum-Creapeau88]

Completeness: \( \forall x \in L \quad \Pr[(P, V)(x) = 1] \geq \frac{2}{3} \)

Soundness: \( \forall x \notin L, \forall PPT \ P^* \quad \Pr[(P^*, V)(x) = 1] \leq \frac{1}{3} \)
So Far...

- **Constructed ZK proofs for all of NP**
  - using commitment schemes

- **Constructed commitment scheme**
  - Based on injective OWF:
    - computationally hiding, perfectly binding
  - Based on Discrete Log:
    - perfectly hiding, computationally binding
Interactive Proofs are More Efficient!

[Lund-Fortnow-Karloff-Nissan90, Shamir90]

Example: Chess
Interactive Proofs are More Efficient!
[Lund-Fortnow-Karloff-Nissan90, Shamir90]

correctness of any computation can be proved:

\[
\text{Time to verify} \approx \text{Space required to do the computation}
\]

\[
\text{Interactive Proof}
\]

\[
\text{IP} = \text{PSPACE}
\]
Interactive Proofs are More Efficient!
[Lund-Fortnow-Karloff-Nissan90, Shamir90]

correctness of any computation can be proved:

\[
\text{Time to verify} \approx \text{Space required to do the computation}
\]

Succinct space $\rightarrow$ succinct interactive proof
Multi-Prover Interactive Proofs

[BenOr-Goldwasser-Kilian-Wigderson88]

\[ P_1 \xrightarrow{a_1} q_1 \xleftarrow{q_2} a_2 \xrightarrow{P_2} V \]

\[ \forall f \text{ computable in time } T: \]
2-provers can convince verifier that \( f(x) = y \), where the runtime of the verifier is only \( |x| \cdot \text{polylog}(T) \) and the communication is \( \text{polylog}(T) \)

motivated by constructing perfect ZK proofs
[Fortnow-Rompel-Sipser88]:

\[ V \]

\[ a_1 \ a_2 \ a_3 \ a_4 \]

\[ a_1 \ a_2 \ a_3 \ a_4 \]

\[ P_1 \]

\[ P_2 \]

\[ q_1 \]

\[ q_2 \]

\[ a_1 \]

\[ a_2 \]

\[ V \]
Probabilistically Checkable Proofs


Read only 3 bits of the proof, and obtain soundness 1/8
Classical proofs

(Zero-knowledge) Interactive proofs

Multi-prover interactive proofs

Probabilistically checkable proofs (PCPs)

Interactive PCP/Interactive oracle proofs

Fiat-Shamir paradigm

SNARGs
THANK YOU