Focus thus far in the course: protecting communication (e.g., message confidentiality and message integrity) with surprising implications (e.g., DSA/ECDSA signatures based on ZK!)

Remainder of course: protecting computations

Zero-knowledge: a defining idea at the heart of theoretical cryptography

\[ \text{Idea will seem very counter-intuitive, but surprisingly powerful} \]

\[ \text{Shoos away the importance and power of definitions (e.g., "What does it mean to know something?")} \]

We begin by introducing the notion of a "proof system"

\[ \text{Goal: A prover wants to convince a verifier that some statement is true} \]

\[ \text{e.g., "This Sudoku puzzle has a unique solution"} \]

\[ \text{The number } N \text{ is a product of two prime numbers } p \text{ and } q \]

\[ \text{I know the discrete log of } h \text{ base } g \]

We model this as follows:

\[ \text{prover}(X) \xrightarrow{\pi} \text{ verifier}(X) \]

\[ X: \text{Statement that the prover is trying to prove (known to both prover and verifier)} \]

\[ \pi: \text{the proof of } X \]

\[ \left\{ \begin{array}{l}
\text{We will write } L \text{ to denote the set of true statements (called a language)}
\end{array} \right. \]

\[ \left\{ \begin{array}{l}
b \in \{0,1\}^* \Rightarrow \text{given statement } X \text{ and proof } \pi \text{, verifier decides whether to accept or reject}
\end{array} \right. \]

Properties we care about:

\[ \text{Completeness: Honest prover should be able to convince honest verifier of true statements} \]

\[ \forall X \in L : \Pr[\pi \leftarrow P(X) : V(X, \pi) = 1] = 1 \]

\[ \text{Could relax requirement to allow for some error} \]

\[ \text{Soundness: Dishonest prover cannot convince honest verifier of false statement} \]

\[ \forall X \notin L : \Pr[\pi \leftarrow P(X) : V(X, \pi) = 1] < \frac{1}{3} \]

\[ \text{Important: We are not restricting to efficient provers (for now)} \]

Typically, proofs are "one-shot" (i.e., single message from prover to verifier) and the verifier's decision algorithm is deterministic

Languages with these types of proof systems precisely coincide with NP (proof of statement \( X \) is to send NP witness \( w \))

Recall that NP is the class of languages where there is a deterministic solution-checker:

\[ L \in \text{NP} \iff \exists \text{ efficiently-computable relation } R \text{ s.t.} \]

\[ \forall X \in L : \exists \omega \in \{0,1\}^{|X|} : R(X, \omega) = 1 \]

\[ \text{Statement \ language \ witness \ \ NP \ relation} \]

Proof system for NP:

\[ \text{prover}(X) \xrightarrow{\omega} \text{ verifier}(X) \]

\[ \left\{ \begin{array}{l}
\text{Accept if } R(X, \omega) = 1
\end{array} \right. \]

Perfect completeness + soundness
Interactive proof systems (Goldwasser–Micali–Rackoff):

\[
\begin{array}{c}
\text{prover} (x) \\
\vdots \\
\end{array}
\rightarrow
\begin{array}{c}
\text{interactive proof} \\
\text{may be inefficient} \\
\end{array}
\rightarrow
\begin{array}{c}
\text{efficient and} \\
\text{randomized} \\
\end{array}
\rightarrow
\begin{array}{c}
\text{verifier} (x) \\
\end{array}
\rightarrow
\begin{array}{c}
\text{Verifier randomness is critical. Otherwise, class of languages that} \\
\text{can be recognized collapses to NP. (See [HRS].)} \\
\end{array}
\]

Interactive proof should satisfy completeness + soundness (as defined earlier).

We define IP[k] to denote class of languages where there is an interactive proof with k messages.

We write IP = IP[pol(n)] where n is the statement length.

(i.e., IP is the class of languages with an interactive proof with polynomially-many rounds)

\[
\text{Verdict (prover) } \rightarrow \text{Verdict (verifier)}
\]

Special case: Arthur–Merlin proofs: verifier randomness is public and known to the prover.

\[
\text{AM}[k] : \text{AM proof with k messages, class AM = AM}[2] \text{ (two-message public-coin proofs)}
\]

For constant k, AM[k] = AM = AM[1] (constant message = 2 message) \rightarrow equivalent to BP-NP (class of languages with randomized reduction from 3-SAT)

\[
B \subseteq C \iff \exists \text{efficient } M : \forall x : Pr[M(A(s)) = B(x)] \geq \frac{1}{2}
\]

Theorem (Goldwasser–Sipser): For every k \in \mathbb{N}, IP[k] \subseteq AM[k+2]

(Any private-coin interactive proof can be simulated by a public-coin interactive proof with two extra rounds)

What is the power of IP?

- For constant number of messages, seems comparable to NP (IP[1] collapses to AM for constant k \in \mathbb{N})
- Going from constant to polynomial number of rounds is significant!

Theorem. (Lund–Fortnow–Karloff–Nisan’90, Shamir’90) IP = PSPACE.

Proof (Idea). We will prove a weaker statement which illustrates all of the main techniques of the proof.

Let 3col be the graph 3-coloring problem

- Given graph $G = (V, E)$, can we color the nodes so two adjacent nodes have different colors? [NP complete]

Let #3col be the problem of counting the number of 3-colorings of a graph.

We will show #3col \in IP (this implies for instance that coNP \subseteq IP since #3col is coNP-hard)

\[
\Rightarrow \#3col \in \#P \text{- complete} \quad (\text{Toda's Theorem: } \text{PH} \subseteq \#P) \\
\Rightarrow \text{counting the number of witnesses to a polynomial-time relation}
\]

Step 1 (Arithmeticization): We will construct a polynomial Pb that outputs 1 on a valid coloring and 0 otherwise.

- Let $G = (V, E)$ be the graph. For each vertex $u \in V$, let $x_u \in \{0, 1, 2\}$ be the associated color.

\[
|V| = n \quad |E| = m
\]
Consider the polynomial
\[ \hat{P}_0(x_1, \ldots, x_n) = \prod_{(u, v) \in E} (x_u - x_v) \]

Suppose \((x_1, \ldots, x_n)\) is an invalid coloring. Then, for some \((u, v) \in E\), \(x_u = x_v\), and \(P_0(x_1, \ldots, x_n) = 0\).

Suppose \((x_1, \ldots, x_n)\) is a valid coloring. Then, for all \((u, v) \in E\), \(x_u - x_v \in \{-2, -1, 1, 2\}\).

Define \(f: \mathbb{R} \to \mathbb{R}\) be a polynomial where \(f(0) = 0\) and \(f(\pm 2) = f(\pm 1) = f(3) = 1\).

For example, \(f(x) = \frac{5}{4} x^2 - \frac{1}{4} x^4\) satisfies the desired properties.

Define \(P_0(x_1, \ldots, x_n) = \prod_{(u, v) \in E} f(x_u - x_v)\)

- For an invalid coloring: \(P_0(x_1, \ldots, x_n) = 0\)
- For a valid coloring: \(P_0(x_1, \ldots, x_n) = 1\)

**Goal:** interactive proof to check sum of this polynomial
\[ K = \sum_{x_1 \in \{0, 1\}} \cdots \sum_{x_n \in \{0, 1\}} P_0(x_1, \ldots, x_n) \]

**Step 2 (Sumcheck protocol):** Instead of working over \(\mathbb{R}\), we will work over \(\mathbb{Z}_p\) (for prime \(p\))

If \(p > 3^n\), this is guaranteed to be correct.

**Approach:** Prover first computes polynomial
\[ P_t(x) = \sum_{x_1 \in \{0, 1\}} \cdots \sum_{x_n \in \{0, 1\}} f(x_u - x_v) \]

- This is a polynomial with degree \(d \leq 4m\), since \(\deg(f) = 4\)
- Polynomial is univariate, so can be described by at most \(4m + 1\) coefficients
- Prover can send \(P_t\) to verifier \((4m + 1)\) coefficients) and verifier can check that
  \[ K = \sum_{x_1 \in \{0, 1\}} P_t(x) \]

This can be checked efficiently! But what if prover cheats and sends \(\hat{P}_t\) that does not satisfy (\(x\))

- Verifier needs to check validity of \(\hat{P}_t\).

**Idea:** Sample \(r \in Z_p\) and ask prover to prove that
\[ \hat{P}_t(r) = \sum_{x_1 \in \{0, 1\}} \cdots \sum_{x_n \in \{0, 1\}} P_0(r, x_1, \ldots, x_n) \]

The verification process can be continued until we get a univariate polynomial.
if the statement is false \( \Rightarrow \) verifier always rejects

otherwise, verifier always accepts

Can now argue soundness inductively:

- for false statement on \( n \) variables: verifier rejects false statement w.p. at least \( (1 - \frac{4n}{p}) \)
- trivial for case where \( n = 1 \)
- for general case on \( n \) variables: \( \exists_{p \leq Z_p} \text{ is not a root of } P_n - P_n \) w.p. \( 1 - \frac{4n}{p} \), in which case prover must show false statement on \( n-1 \) variables

\( \Rightarrow \) soundness \( (1 - \frac{4n}{p})(1 - \frac{4n}{p})^{n-1} = (1 - \frac{4n}{p})^n \)

Choose \( p > 4^{4n} \) so soundness holds with constant prob.

Boolean formula:

\[
\forall x_1, \exists x_2, \forall x_3, \ldots, \exists x_n \Phi(x_1, \ldots, x_n)
\]

Implication: \( \#3SAT \in \text{IP} \Rightarrow \text{coNP} \subseteq \text{IP} \).

Approach directly generalizes to the total quantifier Boolean formula (TQBF) problem which is complete for PSPACE

\( \Rightarrow \) PSPACE \( \subseteq \text{IP} \Rightarrow \text{IP} = \text{PSPACE} \)

\( \Rightarrow \) arithmetic (with linearization), followed by sumcheck

Sumcheck protocols very useful for verifying polynomial-time computations with small communication

"interactive proofs for muggles" [Goldwasser-Kalai-Rothblum '08]

] key building block for succinct arguments and verifiable computation