Succinct Proofs
and Polynomial Commitments
Last Time:
Fiat-Shamir Heuristic
Replacing $V$ with $H$ in a public-coin protocol yields a non-interactive (RO) proof

This Time:
1. Succinct Proofs
2. Succinctness from the PCP Theorem
   (and merkle trees)
3. Polynomial Commitments
Succinct Proofs: Motivation

- Verifiable outsourcing

Alice \[\xrightarrow{f, x} y, \pi\] untrusted
\[\xrightarrow{\text{Verify}(f, x, y, \pi)} 0.13\]
\[\xrightarrow{\text{Pi} \leftarrow \text{Prove}(f, x, y)} \text{not too slow}\]
\[O(\log(x+y))\]

Amazon

\[\xrightarrow{\text{y} \leftarrow f(x)} \text{short!} \leq O(1)\]
Formalizing Succinctness: SNARGs

- \( R(x, w) \) an NP-relation
- Verifier knows \( x \)
- Prover knows \( x, w \)
- Two algs:
  - Prove \( (R, x, w) \rightarrow \Pi \)
  - Verify \( (R, x, \Pi) \rightarrow \{0, 1\} \)
- Succinctness
  - \( |\Pi| \) and Verify are \( o(1w1) \)
- Non-interactive (e.g. via Fiat-Shamir)
- Argument (not proof!)
  - sound only for poly-time \( P^* \)
- "SNARG"
Extensions to SNARGs

- Can add zero knowledge
- Can add knowledge-soundness
- Together, these give a "zk-SNARK"

- Today, we'll stick to SNARGs
- Really, SARGs (interactive)
  - Fiat-Shamir yields a SNARG
How to build a SNARG?

Strawman: $V_R$ - the NP-verifier

$\Pi = \omega \leftarrow \text{too big}$

$\text{Verify}(R, x, \Pi) = V_R(x, \Pi)$

$\implies O(1 \times |1| + 1|w|)$ time, just to read the input.

Necessary? $\implies$

Can we "spot check" proofs?

Amazingly, YES!
The PCP Theorem

"Probabilistically Checkable Proofs"

For any $R(x, w) \in \text{NP}$ there are

- $\text{PCP. Prover}_R(x, w) \rightarrow \pi \in \{0, 1\}^{\text{poly}(|x|)}$
- $\text{PCP. Verifier}_R(x, \pi) \rightarrow \{0, 1\}$

which are complete & sound. Furthermore,

$\text{PCP. Verifier}_R(x, \pi)$ reads only $3$ (random) bits of $\pi$ and runs in $O(|x|)$ time.
PCP story

- Complex proof
  - Multiple papers (80s, 90s)
    - Credits to: Arora, Babai, Feige, Goldwasser, Lovasz, Lund, Motwani, Rackoff, Safra, ...
  - Won the Gödel Prize
  - OCD bits
    - 3 bits
      - Won the Gödel Prize, again
  - Improvement [Johan Håstad '01]
    - 3 bits
      - Won the Gödel Prize, again
  - Simpler proof [Irit Dinur '07]
    - Based on expander graphs
    - Only ?? 44 pages
    - Won the Gödel Prize again
  - Applications to hardness of approximation
- We'll just use it...
PCP + Merkle Tree = SNARG

[Kilian '92] [Micali '94]

Prover\((x, w)\)

\[
\Pi \leftarrow \text{PCP. Prove}_R(x, w)
\]

C ← Merkle. Commit(\(\Pi\))

\[
\text{for } j \in \{1, 2, 3\} \quad P_j \leftarrow \text{Merkle.Open}(\Pi, i_j)
\]

\[
X_j \leftarrow \Pi[ i_j ]
\]

\[
3 \times \text{MerkleVerify}(x_j, P_j) \rightarrow \text{PCP.Verify}
\]

Verifier\((x)\)
Complete? Perfectly!

Sound? Yes, because (informal)
→ PCP is sound
→ Merkle openings are correct (assuming collision resistance)

Zero Knowledge?
→ No (U sees 3 bits)
→ but 2K can be added [Kilian ’97]

Succinct?
→ Yes: \( O(\log_{2}(w)) \) communication
Practical?

→ No

Merkle Trees + PCP Theorem
• very practical
• used frequently

• theoretical
• unimplemented
• big constants

Observation: Our two subsystems are unbalanced...

Idea: Use a weaker proof system (PCP) and a stronger commitment scheme
Polynomial Commitments

- Vector commitments open to elements
- Polynomial commitments open to evaluations

Prover (polynomial f)

\[ c = \text{Commit}(f) \]

Verifier ( )

\[ (x, p(x)) \]

\[ x, \in \mathbb{R} \]

\[ y_i = f(x_i) \]

\[ \Pi_i = \text{Open}(f, x_i) \]

\[ \text{Check}(\Pi_i, c, x_i, y_i) \]
In more detail, a polynomial commitment scheme has 4 algorithms:

- **Setup** \((d) \rightarrow pp\)
- **Commit** \((pp, f) \rightarrow c\)
- **Open** \((pp, f, x) \rightarrow \pi\)
- **Check** \((pp, c, x, y, \pi) \rightarrow \{0, 1\}\)

**Correctness:**

\(\forall d \forall f. \ \deg(f) \leq d \ \forall x \in \text{GF}\)

\[
\Pr \left[ \left( \begin{array}{c}
pp \leftarrow \text{Setup}(d) \\
c \leftarrow \text{Commit}(pp, f) \\
\pi \leftarrow \text{Open}(pp, f, x) \\
y \leftarrow f(x)
\end{array} \right) \right| \right. \\
\left. \left( c, x, y, \pi \right) \rightarrow \{0, 1\} \right] = 1.
\]
Evaluation Binding

\[ \forall \forall A PPT A \]

\[ p_p \leftarrow \text{setup}(d) \]

\[ \Pr \left\{ (f, c, x, y, y', \pi, \pi') \leftarrow A(d, p) \right\} \]

\[ = 1 = \text{Check}(p_p, c, x, y, \pi) \]

\[ \land 1 = \text{Check}(p_p, c, x, y', \pi') \]

is negligible in \( \lambda \)

There are various "knowledge" security definitions too

- e.g. "a prover who creates valid openings must "know" a matching polynomial"
- important, but complex
- we'll ignore them.
Polynomial Review

A polynomial \( p : \mathbb{F} \to \mathbb{F} \) is a function defined by

\[
p(x) = \sum_{i=0}^{d} c_i x^i \quad c_i \in \mathbb{F}
\]

A polynomial's degree is its largest non-zero coefficient.

Polynomials support

- (t) (-) (x)
- quotient-remainder division
  \[
f(x) = q(x)d(x) + r(x) >
\]
  \[\deg(r) < \deg(d)\]
  - because they're a "Euclidean domain"
- A polynomial's roots are the inputs where it is zero
Notation: \( \mathbb{F}^d \) set of polynomials from \( \mathbb{F} \) with degree \( d \) max degree

Polynomial Facts:

- for all \( f \in \mathbb{F}^d \) \( \mathbb{F}[x] \),

\[
\Pr[x \in \mathbb{F} : f(x) = 0] = \frac{d}{|\mathbb{F}|}
\]

- \( f(x) \) is divisible by \( x-a \) iff it has a root at \( a \).

\[
\Rightarrow f(a) = y \quad \text{iff} \quad f(x) - y \quad \text{is divisible by} \quad x-a
\]
Polynomial Commitments from pairings

"KZG" construction: [Kate, Zaverucha, Goldberg '10]

Given \( e: G_1 \times G_2 \to G_T \) with generators \( g_1, g_2 \) and orders \( p \).

Setup \((d)\): \( \alpha \in \mathbb{F}_p \)

\[
pp \leftarrow ( (g_1^{\alpha^i}, \ldots, g_1^{\alpha^d}, g_2^{\alpha}) )
\]

Commit \((pp, f)\):

\[
c \leftarrow g_1^{f(\alpha)} = \prod_{i=0}^{d} (g_1^{\alpha^i})^{f_i}
\]

Open \((pp, f, x)\):

\[
f(x) = \frac{f(x) - f(c)}{x - x} \left( \sum_{i=0}^{d-1} (g_1^{\alpha^i})^{f'_i} \right)
\]

Check \((pp, c, x, y, \pi)\):

\[
e(\pi, g_2^x / g_2^y) = e(c / g_1, y, g_1)
\]
Correctness
\[
\text{e}(\pi, \ g_2^{\alpha-x}) = \text{e}(g_1, f(\alpha)-y, g_2)
\]
\[
f'(\alpha) (\alpha-x) = f(\alpha)-y
\]
\[
f'(\alpha) = \frac{f(\alpha)-y}{\alpha-x}
\]
\[
\checkmark, \text{ by defn.}
\]

Evaluation Binding
Assuming "t-SDH", KZG’10 is eval. binding, for degree bound t.

\[\text{t-SDH: Given } (g, g^{x}, g^{\alpha^2}, \ldots g^{\alpha^t}, g_2, g_1)\]

it is hard to find \[y, g^{\frac{1}{\alpha-x-y}}\]

Proof: Suppose A is an index-binding adv. For KZG’10, we build B, a \[t-\text{SDH} \text{ adv.}\]
same as \text{prep}
\begin{align*}
C, X, Y, \pi, y, \pi' & \rightarrow y \neq y' \\
\text{now what?}
\end{align*}
\begin{align*}
e(\pi, g_2^X/g_2^x) &= e(c/g, y, q_1) \\
e(\pi', g_2^X/g_2^x) &= e(c/g, y', q_1)
\end{align*}
\begin{align*}
\pi^{a-x} &= c/g, y \\
\pi'^{a-x} &= c/g, y'
\end{align*}
\begin{align*}
(\pi/\pi')^{a-x} &= g, y' - y \\
h &= \left(\frac{\pi}{\pi'}\right)^{y' - y} = g, \frac{1}{a-x}
\end{align*}
\begin{align*}
\text{computable, since } y \neq y' & \Rightarrow (x, h)
\end{align*}
\text{t-SDH break.}
Knowledge soundness provable in the algebraic group model, see "Marlin" Chiesa, Hu, Malley, Mishra, Verely, Ward '19
"Sonic" Maller, Bone, Kohlweiss, Meiklejohn '19

→ requires slight changes to scheme.
→ see "Marlin" for details
Problem: Trusted Setup

A trusted setup is a setup procedure whose randomness must be discarded for security.

- controversial! (who runs it?)
- Motivates secure multi-party computation (MPC)
- See Radiolab's "The Ceremony" for the story behind running a trusted setup
Recap:

1. Succinct Arguments
   \[ \Pi_1 = o(1\omega) \]

2. Succinctness from the PCP theorem
   \[ \text{cool, but impractical} \]

3. Polynomial Commitments
   \[ \text{Hope: stronger commitment, more practical SNARK} \]

Next Time:

Succinctness from polynomial commitments