Components of recent universal SNARKs

Ariel Gabizon

Protocol Labs

30 seconds of philosophy: SNARKs and the meaning of life

As the world gets increasingly distributed and digital, SNARKs help us "keep a grip on reality", by ensuring the data we receive is anchored in reality.

In this context, they are vast generalization of hashes and digital signatures.

Evaluating prover polynomials succinctly

Since the beginning of time (LFKN, 1990) humanity has been trying to verify prover polynomial evaluations.

Evaluating prover polynomials succinctly

Verifer: - “Choose a degree 10 polynomial f and keep it in your head”
Prover: - “OK”
Verifer: - “What is $f(7)$?”
Prover: - “10”

Evaluating prover polynomials succinctly

The hard-working honest way - Low degree testing [BFL91,...] (also PCPPs/IOPPs)

Evaluating prover polynomials succinctly

The hard-working honest way - Low degree testing/PCPP/IOPPs

The correct* way - Polynomial commitment scheme (with SRS).

*The opinions expressed in this presentation are objective reality

Notation: "an encoding of x"

$$[x] := x \cdot g = g + \ldots + g \text{ (x times).}$$

Notation: "an encoding of x"

$$[x] := x \cdot \mathbf{g} = \mathbf{g} + \ldots + \mathbf{g} \ (x \text{ times}).$$

You can imagine it means $\mathbf{g}^x$, if it helps you sleep better.

The KZG polynomial commitment scheme

SRS: $[1]$,$[x]$,$\ldots$, $\left[x^{d}\right]$, for random $x \in \mathbb{F}$.

$f(X) = \sum_{i=0}^{d} a_i X^i$

$\mathrm{cm}(f) := \sum_{i=0}^{d} a_i \left[x^i\right] = [f(x)]$

SRS: $[1]$,$[x]$,$\ldots$,$\left[x^{d}\right]$,
for random $x \in \mathbb{F}$.

$\mathsf{cm}(f) := [f(x)]$

$\mathsf{open}(f, i) := [h(x)]$, where $h(X) := \frac{f(X) - f(i)}{X - i}$

$\mathsf{cm}(\mathbf{f}) := [\mathbf{f}(x)]$

$\mathsf{open}(\mathbf{f}, \mathbf{i}) := [\mathbf{h}(x)]$, where $\mathbf{h}(\mathbf{X}) := \frac{\mathbf{f}(\mathbf{X}) - \mathbf{f}(\mathbf{i})}{\mathbf{X} - \mathbf{i}}$

exercise - using pairing, define:

$\mathsf{verify}(\mathsf{cm}, \boldsymbol{\pi}, z, \mathbf{i})$, where z is allegedly $\mathbf{f}(\mathbf{i})$

$\mathsf{cm}(\mathbf{f}) := [\mathbf{f}(x)]$

$\mathsf{open}(\mathbf{f}, \mathbf{i}) := [\mathbf{h}(x)]$, where $\mathbf{h}(\mathbf{X}) := \frac{\mathbf{f}(\mathbf{X}) - \mathbf{f}(\mathbf{i})}{\mathbf{X} - \mathbf{i}}$

$\mathsf{verify}(\mathsf{cm}, \boldsymbol{\pi}, z, \mathbf{i})$:

$$e(\mathsf{cm} - [z], [1]) \stackrel{?}{=} e(\boldsymbol{\pi}, [x - \mathbf{i}])$$

$\mathsf{cm}(\mathbf{f}) := [\mathbf{f}(x)]$

$\mathsf{open}(\mathbf{f}, \mathbf{i}) := [\mathbf{h}(x)]$, where $\mathbf{h}(\mathbf{X}) := \frac{\mathbf{f}(\mathbf{X}) - \mathbf{f}(\mathbf{i})}{\mathbf{X} - \mathbf{i}}$

$\mathsf{verify}(\mathsf{cm}, \boldsymbol{\pi}, z, \mathbf{i})$:

$$e(\mathsf{cm} - [z], [1]) \stackrel{?}{=} e(\boldsymbol{\pi}, [x - \mathbf{i}])$$

Thm[KZG,MBKM]: *This works.*

Application: Proof of retrievability

Application: Proof of retrievability

Server: Commit to file as coefficients $\{a_0, \ldots, a_d\}$ of polynomial f.

Client: Time-to-time query f at a random point r.

Application: Proof of retrievability

Server: Commit to file as coefficients $\{a_0, \ldots, a_d\}$ of polynomial f.

Client: Time-to-time query f at a random point r.

Probability of server computing $f(r)$ correctly w/o remembering all coeffs of f is negligible.

Application: The Sonic helper [MBKM19]

Bi-variate $S(X, Y)$, evaluation points $\{(x_j, y_j)\}_{j \in \{1..t\}}$, values $\{z_j\}$

Application: The Sonic helper [MBKM19]

Bi-variate $S(X, Y)$, evaluation points $\{(x_j, y_j)\}_{j\in\{1..t\}}$, values $\{z_j\}$

Helper $\mathcal{H}$ wants to convince verifier $\mathcal{V}$ that $\forall j \in \{1..t\}$, $S(x_j, y_j) = z_j$.

Application: The Sonic helper [MBKM19]

Bi-variate $S(X, Y)$, evaluation points $\{(x_j, y_j)\}_{j \in \{1..t\}}$, values $\{z_j\}$

Helper $\mathcal{H}$ wants to convince verifier $\mathcal{V}$ that $\forall j \in \{1..t\}$, $S(x_j, y_j) = z_j$.

$\mathcal{V}$'s work: only *one* evaluation of S!

The Sonic helper [MBKM19]

1. $\forall \mathbf{j}$, $\mathcal{H}$ sends $\mathcal{S}_{\mathbf{j}} := \mathsf{cm}(\mathcal{S}(\mathbf{X}, \mathbf{y}_{\mathbf{j}}))$.

The Sonic helper [MBKM19]

1. $\forall \mathbf{j}$, $\mathcal{H}$ sends $\mathcal{S}_j := \mathrm{cm}(\mathcal{S}(X, y_j))$.

If $\mathcal{H}$ would convince $\mathcal{V}$ $\mathcal{S}_j$'s are correct, he could just open them at x_j.

The Sonic helper [MBKM19]

1. $\forall j$, $\mathcal{H}$ sends $\mathcal{S}_j := \mathsf{cm}(\mathcal{S}(X, y_j))$.
2. $\mathcal{V}$ chooses random $u \in \mathbb{F}$.
3. $\mathcal{H}$ sends $C := \mathsf{cm}(\mathcal{S}(u, Y))$.

The Sonic helper [MBKM19]

1. $\forall \mathbf{j}$, $\mathcal{H}$ sends $\mathcal{S}_{\mathbf{j}} := \mathsf{cm}(\mathcal{S}(\mathbf{X}, \mathbf{y}_{\mathbf{j}}))$.
2. $\mathcal{V}$ chooses random $\mathbf{u} \in \mathbb{F}$.
3. $\mathcal{H}$ sends $\mathbf{C} := \mathsf{cm}(\mathcal{S}(\mathbf{u}, \mathbf{Y}))$.
4. $\forall \mathbf{j}$, $\mathcal{H}$ opens $\mathbf{C}$ at $\mathbf{y}_{\mathbf{j}}$ and $\mathcal{S}_{\mathbf{j}}$ at $\mathbf{u}$. $\mathcal{V}$ checks they open to same value.

At this point $\mathcal{V}$ knows $\mathbf{S}_{\mathbf{j}}$'s are correct **if** $\mathbf{C}$ *is correct.*

The Sonic helper [MBKM19]

1. $\forall \mathbf{j}$, $\mathcal{H}$ sends $\mathcal{S}_j := \mathrm{cm}(\mathcal{S}(\mathbf{X}, y_j))$.
2. $\mathcal{V}$ chooses random $u \in \mathbb{F}$.
3. $\mathcal{H}$ sends $\mathbf{C} := \mathrm{cm}(\mathcal{S}(u, \mathbf{Y}))$.
4. $\forall \mathbf{j}$, $\mathcal{H}$ opens $\mathbf{C}$ at y_j and $\mathcal{S}_j$ at x_j. $\mathcal{V}$ checks they open to same value.
5. $\mathcal{V}$ chooses random $v \in \mathbb{F}$ and computes $s(u, v)$.

The Sonic helper [MBKM19]

1. $\forall j$, $\mathcal{H}$ sends $\mathcal{S}_j := \mathrm{cm}(\mathcal{S}(X, y_j))$.
2. $\mathcal{V}$ chooses random $u \in \mathbb{F}$.
3. $\mathcal{H}$ sends $C := \mathrm{cm}(\mathcal{S}(u, Y))$.
4. $\forall j$, $\mathcal{H}$ opens C at y_j and $\mathcal{S}_j$ at x_j. $\mathcal{V}$ checks they open to same value.
5. $\mathcal{V}$ chooses random $v \in \mathbb{F}$ and computes $s(u, v)$.
6. $\mathcal{H}$ opens C at v, and $\mathcal{V}$ checks it opens to $s(u, v)$.

3rd application: sumchecks

3rd application: sumchecks

$\mathcal{P}$ has polynomial $\mathbf{f}$. $\mathbf{H} \subset \mathbb{F}$.

$\mathcal{V}$ wants to check:

$$\sum_{x \in \mathbf{H}} \mathbf{f}(x) = 0$$

The Aurora trick for sumcheck [BCRSV19]

Lemma

When $\mathbf{H} \subset \mathbb{F}$ *is a multiplicative subgroup of size* $\mathbf{n}$*, and* $\mathbf{deg}(\mathbf{g}) < \mathbf{n}$

$$\sum_{x \in \mathbf{H}} \mathbf{g}(x) = 0$$

iff $\mathbf{g}$ *has constant coefficient* 0*.*

The Aurora trick for sumcheck [BCRSV19]

$Z_H := \prod_{x \in H}(X - x)$

By lemma suffices to show g_1, g_2,
$\deg(g_2) < n - 1$ such that

$$f(X) \equiv g_1(X) \cdot Z_H(X) + X \cdot g_2(X)$$

AuroraLight [G19]

Check $\mathbf{f}$ has this form by sending commitments to $\mathbf{g}_1, \mathbf{g}_2$, and openning $\mathbf{f}, \mathbf{g}_1, \mathbf{g}_2$ at random point.

AuroraLight [G19]

Check $\mathbf{f}$ has this form by sending commitments to $\mathbf{g}_1, \mathbf{g}_2$, and openning $\mathbf{f}, \mathbf{g}_1, \mathbf{g}_2$ at random point.

Using Aurora+Sonic ideas:

Corollary

Universal SRS SNARK with prover almost as fast as [Groth16] (But longer proofs than Sonic, and no fully succinct mode).