Ranged Polynomial Protocols

Ariel Gabizon

Aztec

Outline

- A few slides of motivation and context
- Polynomial Protocols - dfns,results + open question.

Succinct arguments in a nutshell

Public program $\mathbf{T}$, public output z.

Succinct arguments in a nutshell

Public program $\mathbf{T}$, public output z.

Want to prove "I know input x for program $\mathbf{T}$ that generates output z.

Succinct arguments in a nutshell

Public program $\mathbf{T}$, public output z.

Want to prove "I know input x for program $\mathbf{T}$ that generates output z.

Want proof size and verification time to be much smaller than run time of $\mathbf{T}$.

(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Succinct arguments in a nutshell

Public program $\mathbf{T}$, public output z.

Want to prove "I know input x for program $\mathbf{T}$ that generates output z.

Want proof size and verification time to be much smaller than run time of $\mathbf{T}$.
(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Arithmeitization [LFKN,......]: Reduce claim to claim of form "I know polynomials that satisfy some identity"

Succinct arguments in a nutshell

Public program $\mathbf{T}$, public output z.

Want to prove "I know input x for program $\mathbf{T}$ that generates output z.

Want proof size and verification time to be much smaller than run time of $\mathbf{T}$.
(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Arithmeitization [LFKN,.....]: Reduce claim to claim of form "I know polynomials that satisfy some identity"

Succinct arguments in a nutshell

Advantage of claims about polynomials is that suffice to check at one random point

Succinct arguments in a nutshell

Advantage of claims about polynomials is that suffice to check at one random point

But need to solve "chicken and egg problem": Prover must commit to polynomials before knowing the challenge point.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment cm$(\mathbf{f})$ to polynomial.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathrm{cm}(\mathbf{f})$ to polynomial.
- Later Verifier can choose value $\mathbf{i} \in \mathbb{F}$.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathsf{cm}(\mathbf{f})$ to polynomial.
- Later Verifier can choose value $\mathbf{i} \in \mathbb{F}$.
- Prover sends back $z = \mathbf{f}(\mathbf{i})$; together with proof $\mathsf{open}(\mathbf{f}, \mathbf{i})$ that z is correct.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathrm{cm}(\mathbf{f})$ to polynomial.
- Later Verifier can choose value $\mathbf{i} \in \mathbb{F}$.
- Prover sends back $\mathbf{z} = \mathbf{f}(\mathbf{i})$; together with proof $\mathrm{open}(\mathbf{f}, \mathbf{i})$ that $\mathbf{z}$ is correct.

KZG give us PCS with commitments and openings are practically 32 bytes.
Notation: $[x] = \mathbf{g}^x$ where $\mathbf{g}$ generator of elliptic curve group.

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

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

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

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

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

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

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

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

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

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

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

Idealized Polynomials Protocols

Preprocessing/inputs: : $\mathcal{P}$ and $\mathcal{V}$ agree in advance on $\mathbf{g}_1, \ldots, \mathbf{g}_\mathbf{t} \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$.

Protocol:

1. $\mathcal{P}$'s msgs are to ideal party $\mathbf{I}$. Must be $\mathbf{f}_\mathbf{i} \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$.
2. At protocol end $\mathcal{V}$ asks $\mathbf{I}$ if some (constant number) of identities hold between $\{\mathbf{f}_1, \ldots, \mathbf{f}_\ell, \mathbf{g}_1, \ldots, \mathbf{g}_\mathbf{t}\}$. Outputs $\mathbf{acc}$ iff they do.

$$\mathfrak{d}(\mathbf{P}) := \left(\sum_{i \in [\ell]} \deg(f_i) + 1 \right)$$

.

[1]similar statements in Marlin/Fractal/Supersonic

$$\mathfrak{d}(\mathbf{P}) := \left(\sum_{\mathbf{i}\in[\ell]} \mathsf{deg}(\mathbf{f_i}) + 1\right)$$

.

Thm:[1] Can compile to "real" protocol in Algebraic Group Model, where prover complexity $\mathfrak{d}(\mathbf{P})$.

[1] similar statements in Marlin/Fractal/Supersonic

$$\mathfrak{d}(\mathbf{P}) := \left(\sum_{\mathbf{i} \in [\ell]} \deg(\mathbf{f_i}) + 1 \right)$$

.

Thm:[1] Can compile to "real" protocol in Algebraic Group Model, where prover complexity $\mathfrak{d}(\mathbf{P})$.

proof sketch: Use [KZG] polynomial commitment scheme. $\mathcal{P}$ commits to all polys. $\mathcal{V}$ checks identity at random challenge point.

---

[1] similar statements in Marlin/Fractal/Supersonic

Ranged polynomials protocols

Preprocessing/inputs: Predefined polynomials
$g_1, \ldots, g_t \in \mathbb{F}_{<d}[X]$
Range: $H \subset \mathbb{F}$.

Ranged polynomials protocols

Preprocessing/inputs: Predefined polynomials $g_1, \ldots, g_t \in \mathbb{F}_{<d}[X]$
Range: $H \subset \mathbb{F}$.

Protocol:

1. $\mathcal{P}$'s msgs are to ideal party I. Must be $f_i \in \mathbb{F}_{<d}[X]$.
2. At end, $\mathcal{V}$ asks I if some identity holds between $\{f_1, \ldots, f_\ell, g_1, \ldots, g_t\}$ **on** H.

$\mathbb{H}$-ranged protocol using polynomial protocol:

$\mathcal{V}$ wants to check identities $\mathbf{P}_1, \mathbf{P}_2$ on $\mathbb{H}$.

- After $\mathcal{P}$ finished sending $\{\mathbf{f_i}\}$, $\mathcal{V}$ sends random $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{F}$.

$\mathbb{H}$-ranged protocol using polynomial protocol:

$\mathcal{V}$ wants to check identities P_1, P_2 on $\mathbb{H}$.

- After $\mathcal{P}$ finished sending $\{f_i\}$, $\mathcal{V}$ sends random $\alpha_1, \alpha_2 \in \mathbb{F}$.
- $\mathcal{P}$ sends $\mathsf{T} \in \mathbb{F}_{<d}[X]$.

$\mathbb{H}$-ranged protocol using polynomial protocol:

$\mathcal{V}$ wants to check identities P_1, P_2 on H.

- After $\mathcal{P}$ finished sending $\{f_i\}$, $\mathcal{V}$ sends random $a_1, a_2 \in \mathbb{F}$.
- $\mathcal{P}$ sends $T \in \mathbb{F}_{<d}[X]$.
- $\mathcal{V}$ checks identity $a_1 \cdot P_1 + a_2 \cdot P_2 \equiv T \cdot Z_H$.

$Z_H(X) := \prod_{a \in H}(X - a)$.
(Z_H will be a preprocessed polynomial).

$\mathbf{H}$-ranged protocol using polynomial protocol:

Motivates - for $\mathbf{H}$-ranged protocol $\mathbf{P}$ define

$$\mathfrak{d}(\mathbf{P}) := \left(\sum_{\mathbf{i}\in[\ell]} \deg(\mathbf{f_i}) + 1\right) + \mathbf{D} - |\mathbf{H}|.$$

$\mathbf{D}$:= max degree of identity $\mathbf{C}$ checked in exec with honest $\mathcal{P}$.

Multiset equality check

Given $\mathbf{a}, \mathbf{b} \in \mathbb{F}^3$, want to check $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\} \stackrel{?}{=} \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$

Multiset equality check

Given $\mathbf{a}, \mathbf{b} \in \mathbb{F}^3$, want to check
$\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\} \stackrel{?}{=} \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$

Choose random $\boldsymbol{\gamma} \in \mathbb{F}$. Check

$$(\mathbf{a}_1+\boldsymbol{\gamma})(\mathbf{a}_2+\boldsymbol{\gamma})(\mathbf{a}_3+\boldsymbol{\gamma}) \stackrel{?}{=} (\mathbf{b}_1+\boldsymbol{\gamma})(\mathbf{b}_2+\boldsymbol{\gamma})(\mathbf{b}_3+\boldsymbol{\gamma})$$

Multiset equality check

Given $\mathbf{a}, \mathbf{b} \in \mathbb{F}^3$, want to check
$\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\} \stackrel{?}{=} \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$

Choose random $\boldsymbol{\gamma} \in \mathbb{F}$. Check

$$(\mathbf{a}_1+\boldsymbol{\gamma})(\mathbf{a}_2+\boldsymbol{\gamma})(\mathbf{a}_3+\boldsymbol{\gamma}) \stackrel{?}{=} (\mathbf{b}_1+\boldsymbol{\gamma})(\mathbf{b}_2+\boldsymbol{\gamma})(\mathbf{b}_3+\boldsymbol{\gamma})$$

If $\mathbf{a}, \mathbf{b}$ different as sets then w.h.p products different.

Multiset equality check

Given $\mathbf{a}, \mathbf{b} \in \mathbb{F}^3$, want to check
$\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\} \stackrel{?}{=} \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$

Choose random $\boldsymbol{\gamma} \in \mathbb{F}$. Check

$$(\mathbf{a}_1 + \boldsymbol{\gamma})(\mathbf{a}_2 + \boldsymbol{\gamma})(\mathbf{a}_3 + \boldsymbol{\gamma}) \stackrel{?}{=} (\mathbf{b}_1 + \boldsymbol{\gamma})(\mathbf{b}_2 + \boldsymbol{\gamma})(\mathbf{b}_3 + \boldsymbol{\gamma})$$

If $\mathbf{a}, \mathbf{b}$ different as sets then w.h.p products different.

Multiset equality check - polynomial version

Given $\mathbf{f}, \mathbf{g} \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$, want to check
$\{\mathbf{f}(x)\}_{x \in H} \stackrel{?}{=} \{\mathbf{g}(x)\}_{x \in H}$ as multisets

Reduces to:

$H = \{\alpha, \alpha^2, \ldots, \alpha^n\}$.

$\mathcal{P}$ has sent $f', g' \in \mathbb{F}_{<n}[X]$.

Wants to prove:

$$\prod_{i \in [n]} f(\alpha^i) = \prod_{i \in [n]} g(\alpha^i)$$

$f := f' + \gamma, g := g' + \gamma$

Multiplicative subgroups:

$\mathbf{H} = \left\{\alpha, \alpha^2, \ldots, \alpha^n = 1\right\}$.

$\mathbf{L}_i$ is i'th lagrange poly of $\mathbf{H}$:

$$\mathbf{L}_i(\alpha^i) = 1, \mathbf{L}_i(\alpha^j) = 0, j \neq i$$

Checking products with $\mathbf{H}$-ranged protocols [GWC19]

1. $\mathcal{P}$ computes Z with $Z(\alpha) = 1, Z(\alpha^i) = \prod_{j<i} f(\alpha^j)/g(\alpha^j)$.
2. Sends Z to $\mathbf{I}$.

Checking products with $\mathbb{H}$-ranged protocols [GWC19]

1. $\mathcal{P}$ computes Z with
   $Z(\alpha) = 1, Z(\alpha^i) = \prod_{j<i} f(\alpha^j)/g(\alpha^j)$.
2. Sends Z to $\mathbf{I}$.
3. $\mathcal{V}$ checks following identities on $\mathbb{H}$.
   3.1 $L_1(X)(Z(X) - 1) = 0$
   3.2 $Z(X)f(X) = Z(\alpha \cdot X)g(X)$

Checking products with $\mathbf{H}$-ranged protocols [GWC19]

1. $\mathcal{P}$ computes Z with
   $Z(\alpha) = 1, Z(\alpha^i) = \prod_{j<i} f(\alpha^j)/g(\alpha^j)$.
2. Sends Z to $\mathbf{I}$.
3. $\mathcal{V}$ checks following identities on $\mathbf{H}$.
   3.1 $L_1(X)(Z(X) - 1) = 0$
   3.2 $Z(X)f(X) = Z(\alpha \cdot X)g(X)$

We get $\mathfrak{d}(P) = n + 2n - |\mathbf{H}| = 2n$.

Example 2: Range checks

Integer $M < n$. Given $f \in \mathbb{F}_{<n}[X]$, want to check $f(x) \in [1..M]$ for each $x \in H$.

Example 2: Range checks

Integer $M < n$. Given $f \in \mathbb{F}_{<n}[X]$, want to check $f(x) \in [1..M]$ for each $x \in H$.
(most?) common SNARK operation: SNARK recursion requires simulating one field using another

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$

Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$
Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$

Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and f.

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$

Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   - 3.1 Mutli-set equality between s and f.
   - 3.2 $s(\alpha) = 1$
   - 3.3 $s(\alpha^n) = M$

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$

Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and f.
   3.2 $s(\alpha) = 1$
   3.3 $s(\alpha^n) = M$
   3.4 For each $x \in H \setminus \{1\}$,

Example 2: Range checks

Simplifying assumption: $[1..M] \subset \{f(x)\}_{x \in H}$
Protocol:

1. $\mathcal{P}$ computes "sorted version of f": $s \in \mathbb{F}_{<n}[X]$ with $\{s(x)\}_{x \in H} = \{f(x)\}_{x \in H}$, $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and f.
   3.2 $s(\alpha) = 1$
   3.3 $s(\alpha^n) = M$
   3.4 For each $x \in H \setminus \{1\}$,

$$(s(x \cdot \alpha) - s(x))^2 = s(x \cdot \alpha) - s(x)$$

We get $\mathfrak{d}(\mathbf{P}) = 3n$

To remove assumption use preprocessed "table poly" $\mathbf{t}$ with $\{\mathbf{t}(x)\}_{x\in H} = [1..M]$
(details on next slide)

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$

Protocol:

1. $\mathcal{P}$ computes "sorted version of $f \cup t$":
   $s \in \mathbb{F}_{<n+M}[X]$ with
   $\{s(x)\}_{x \in H} = \{f(x), t(x)\}_{x \in H}$,
   $s(\alpha^i) \leq s(\alpha^{i+1})$.

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$

Protocol:

1. $\mathcal{P}$ computes "sorted version of $f \cup t$":
   $s \in \mathbb{F}_{<n+M}[X]$ with
   $\{s(x)\}_{x \in H} = \{f(x), t(x)\}_{x \in H}$,
   $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$

Protocol:

1. $\mathcal{P}$ computes "sorted version of $f \cup t$":
   $s \in \mathbb{F}_{<n+M}[X]$ with
   $\{s(x)\}_{x \in H} = \{f(x), t(x)\}_{x \in H}$,
   $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and $f \cup t$.

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$
Protocol:

1. $\mathcal{P}$ computes "sorted version of $f \cup t$":
   $s \in \mathbb{F}_{<n+M}[X]$ with
   $\{s(x)\}_{x \in H} = \{f(x), t(x)\}_{x \in H}$,
   $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and $f \cup t$.
   3.2 $s(\alpha) = 1$
   3.3 $s(\alpha^n) = M$
   3.4 For each $x \in H \setminus \{1\}$,

Preprocessed poly: $t \in \mathbb{F}_{<M}[X]$ with $\{t(x)\}_{x \in H} = [1..M]$

Protocol:

1. $\mathcal{P}$ computes "sorted version of $f \cup t$":
   $s \in \mathbb{F}_{<n+M}[X]$ with
   $\{s(x)\}_{x \in H} = \{f(x), t(x)\}_{x \in H}$,
   $s(\alpha^i) \leq s(\alpha^{i+1})$.
2. $\mathcal{P}$ sends s to $\mathbf{I}$.
3. $\mathcal{V}$ checks that
   3.1 Mutli-set equality between s and $f \cup t$.
   3.2 $s(\alpha) = 1$
   3.3 $s(\alpha^n) = M$
   3.4 For each $x \in H \setminus \{1\}$,
   $$(s(x \cdot \alpha) - s(x))^2 = s(x \cdot \alpha) - s(x)$$

We get
$\mathfrak{d}(P) = \deg(s) + \deg(Z) + D - |H| = 3n + 4M.$

Given integer $\mathbf{d}$ decomposing each element to $\mathbf{d}$ elements in range $\mathbf{M}^{1/\mathbf{d}}$ can give us

$$\mathfrak{d}(\mathbf{P}) = 4\mathbf{dn} + 4\mathbf{M}^{1/\mathbf{d}}$$

(by sending an auxiliary polynomial of degree $< \mathbf{dn}$ with the decomposition of each element and then running the $\mathbf{M}^{1/\mathbf{d}}$ size range proof on this polynomial).

Question: can we do better?