On SNARKs with universal updatable setup

Ariel Gabizon

Aztec Protocol

(preproccessing) zk-SNARKs

Arithmetic circuit $\mathbf{C}$. "Public input" x.

- $\mathcal{P}$ can prove she knows w s.t. $\mathbf{C}(x, w) = 0$.

(preproccessing) zk-SNARKs

Arithmetic circuit $\mathbf{C}$. "Public input" x.

- $\mathcal{P}$ can prove she knows w s.t. $\mathbf{C}(x, w) = 0$.
- Proof size - polylog$|w|$.

(preproccessing) zk-SNARKs

Arithmetic circuit $\mathbf{C}$. “Public input” x.

- $\mathcal{P}$ can prove she knows w s.t. $\mathbf{C}(x, w) = 0$.
- Proof size - polylog$|w|$.
- Proof doesn't leak info on w.

(preproccessing) zk-SNARKs

Arithmetic circuit $\mathbf{C}$. "Public input" x.

- $\mathcal{P}$ can prove she knows w s.t. $\mathbf{C}(x, w) = 0$.
- Proof size - polylog$|w|$.
- Proof doesn't leak info on w.
- One time setup procedure to generate common reference string (depends on $\mathbf{C}$, not on x).

Talk outline

1. The problem with prev constructions.
2. The solution with recent ones.

Talk outline

1. The problem with prev constructions.
2. The solution with recent ones.

We probably won't get too far with 2, unless you want to skip 1.

The QAP approach [GGPR,..]

Reduces to $\mathcal{P}$ knowing deg $< \mathbf{n}$ polynomials $\mathbf{L}, \mathbf{R}, \mathbf{O}$ with

1. $\mathbf{Z} \mid \mathbf{L} \cdot \mathbf{R} - \mathbf{O}$,
2. $(\mathbf{L}, \mathbf{R}, \mathbf{O}) \in \mathbf{V_C}$.

$\mathbf{Z}(\mathbf{X}) := \mathbf{X}^{\mathbf{n}} - 1$. $\mathbf{n}$ = num. of mult gates

$\mathbf{V_C}$:= affine subspace depending on $\mathbf{C}$

Verifying first cond. with pairings+KEA [Groth10,...]

Setup: uniform secret $s \in \mathbb{F}$, $\mathbf{g} \in \mathbf{G}$-group with pairing.
CRS: $\mathbf{g}, \mathbf{g}^s, \ldots, \mathbf{g}^{s^n}$.

Verifying first cond. with pairings+KEA [Groth10,...]

Setup: uniform secret $s \in \mathbb{F}$, $g \in \mathbf{G}$-group with pairing.
CRS: $g, g^s, \ldots, g^{s^n}$.

$\mathcal{P}$ computes $\mathsf{T} = (L \cdot R - O)/Z$.

$\mathcal{P}$ computes and sends $g^{L(s)}, g^{R(s)}, g^{O(s)}, g^{\mathsf{T}(s)}$.

Verifying first cond. with pairings+KEA [Groth10,...]

Setup: uniform secret $s \in \mathbb{F}$, $g \in \mathbf{G}$-group with pairing.
CRS: $g, g^s, \ldots, g^{s^n}$.

$\mathcal{P}$ computes $\mathsf{T} = (\mathbf{L} \cdot \mathbf{R} - \mathbf{O})/\mathsf{Z}$.

$\mathcal{P}$ computes and sends $g^{\mathbf{L}(s)}, g^{\mathbf{R}(s)}, g^{\mathbf{O}(s)}, g^{\mathsf{T}(s)}$.

$\mathcal{V}$ checks using pairings if

$$\mathbf{L}(s) \cdot \mathbf{R}(s) - \mathbf{O}(s) = \mathsf{T}(s) \cdot \mathsf{Z}(s)$$

CRS:=$\mathbf{g}, \mathbf{g}^s, \ldots, \mathbf{g}^{s^n}$.

CRS is universal and updatable:

- Universal - depends only on circuit size

CRS:=$\mathbf{g}, \mathbf{g}^{s}, \ldots, \mathbf{g}^{s^n}$.

CRS is universal and updatable:

- Universal - depends only on circuit size
- Updatable: At any point new party $\mathbf{P}$ can update CRS with new secret $\mathbf{s}'$

$$\mathsf{CRS}_{\mathsf{new}} := \mathbf{g}, (\mathbf{g}^{s})^{s'}, \ldots, (\mathbf{g}^{s^n})^{s'^n}$$

CRS:=$\mathbf{g}, \mathbf{g}^{s}, \ldots, \mathbf{g}^{s^n}$.

CRS is universal and updatable:

- Universal - depends only on circuit size
- Updatable: At any point new party $\mathbf{P}$ can update CRS with new secret $\mathbf{s'}$

$$\text{CRS}_{\text{new}} := \mathbf{g}, (\mathbf{g}^{s})^{s'}, \ldots, (\mathbf{g}^{s^n})^{s'^n}$$

Set of all updaters from all time is required to reconstruct secret of current CRS.

Verifying second condition

Now to check $(\mathbf{L}, \mathbf{R}, \mathbf{O}) \in \mathbf{V_C}$.

Include in CRS $\mathbf{g}^{\boldsymbol{\alpha} \cdot \mathbf{f}(s)}$ for secret $\boldsymbol{\alpha} \in \mathbb{F}$ (only) for $\mathbf{f} \in \mathbf{V_C}$.

Ruins universality and updatability of CRS.

Polynomial commitment schemes

[Groth10,GGPR,..] approach: check equation at secret point in the exponent, *limited to degree two checks because of pairings*

Polynomial commitment schemes

[Groth10,GGPR,..] approach: check equation at secret point in the exponent, *limited to degree two checks because of pairings*

"PCS approach:" [MBKM,..,..] $\mathcal{P}$ will commit to its polynomials and open them later at random verifier point.

Polynomial commitment schemes

[Groth10,GGPR,..] approach: check equation at secret point in the exponent, *limited to degree two checks because of pairings*

"PCS approach:" [MBKM,..,..] $\mathcal{P}$ will commit to its polynomials and open them later at random verifier point.

Can be done with single group element commit/opens using [KZG] scheme.

The KZG polynomial commitment scheme

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

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

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

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

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

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

Idealized Polynomials Protocols

Preprocessing: $\mathcal{V}$ chooses polynomials $\mathbf{g}_1, \ldots, \mathbf{g_t} \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$.

Protocol:

1. $\mathcal{P}$'s msgs are to ideal party $\mathbf{I}$. Must be $\mathbf{f_i} \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$.
2. At protocol end $\mathcal{V}$ asks $\mathbf{I}$ if some identities hold between $\{\mathbf{f}_1, \ldots, \mathbf{f}_\ell, \mathbf{g}_1, \ldots, \mathbf{g_t}\}$. Outputs $\mathbf{acc}$ iff they do.

Plonk [GWC19]:

1. All you need is a permutation check.
2. Permutations are easier to check on mutliplicative subgroups

example: Prove knowledge of a, b, c with

$$(a + b) \cdot c = 7$$

Left values: $\mathbf{l}_1, \mathbf{l}_2$
Right values: $\mathbf{r}_1, \mathbf{r}_2$
Output values: $\mathbf{o}_1, \mathbf{o}_2$

Left values: $\mathbf{l}_1, \mathbf{l}_2$
Right values: $\mathbf{r}_1, \mathbf{r}_2$
Output values: $\mathbf{o}_1, \mathbf{o}_2$

Gate checks: $\mathbf{l}_1 + \mathbf{r}_1 = \mathbf{o}_1, \mathbf{l}_2 \cdot \mathbf{r}_2 = \mathbf{o}_2$
Wire/copy checks: $\mathbf{o}_1 = \mathbf{l}_2$
Public input checks: $\mathbf{o}_2 = 7$.

Copy checks with permutations

similar to [Groth09,BCGGHJ17]

$$\mathbf{V} = (\mathbf{l}_1, \mathbf{l}_2, \mathbf{r}_1, \mathbf{r}_2, \mathbf{o}_1, \mathbf{o}_2)$$

Copy checks with permutations

similar to [Groth09,BCGGHJ17]

$$\mathbf{V} = (\mathbf{l}_1, \mathbf{l}_2, \mathbf{r}_1, \mathbf{r}_2, \mathbf{o}_1, \mathbf{o}_2)$$

$\mathbf{o}_1 = \mathbf{l}_2$ iff $\mathbf{V} = \boldsymbol{\sigma}(\mathbf{V})$
For permutation $\boldsymbol{\sigma} = (25)$

Part 2: Permutations are easier to check on mutliplicative subgroups

$\mathbb{H}$-ranged Polynomials Protocols

Preprocessing: $\mathcal{V}$ chooses polynomials $\mathbf{g}_1, \ldots, \mathbf{g}_t \in \mathbb{F}_{<\mathbf{d}}[\mathbf{X}]$, $\mathbf{H} \subset \mathbb{F}$.

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 end, $\mathcal{V}$ asks $\mathbf{I}$ if some identities hold between $\{\mathbf{f}_1, \ldots, \mathbf{f}_\ell, \mathbf{g}_1, \ldots, \mathbf{g}_\mathbf{t}\}$ **on** $\mathbf{H}$.

Checking permutations with $\mathbf{H}$-ranged protocols

Permutation $\sigma : [n] \to [n]$. $H = \{\alpha, \alpha^2, \ldots, \alpha^n\}$.

$\mathcal{P}$ has sent $f \in \mathbb{F}_{<d}[X]$.

Wants to prove $f = \sigma(f)$:

$$\forall i \in [n], f(\alpha^i) = f(\alpha^{\sigma(i)})$$

Using [BG12] reduces to:

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

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

Wants to prove:

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

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

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

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

1. $\mathcal{P}$ computes $\mathbf{Z}$ with $\mathbf{Z}(\alpha) = 1, \mathbf{Z}(\alpha^{\mathbf{i}}) = \prod_{\mathbf{j}<\mathbf{i}} \mathbf{f}(\alpha^{\mathbf{j}})/\mathbf{g}(\alpha^{\mathbf{j}})$.
2. Sends $\mathbf{Z}$ to $\mathbf{I}$.
3. $\mathcal{V}$ checks following identities on $\mathbf{H}$.
   3.1 $\mathbf{L}_1(\mathbf{X})(\mathbf{Z}(\mathbf{X}) - 1) = 0$
   3.2 $\mathbf{Z}(\mathbf{X})\mathbf{f}(\mathbf{X}) = \mathbf{Z}(\alpha \cdot \mathbf{X})\mathbf{g}(\mathbf{X})$
   3.3 $\mathbf{L}_\mathbf{n}(\mathbf{X})(\mathbf{Z}(\alpha \cdot \mathbf{X}) - 1) = 0$

The bottom line (on BLS-381 curve)

600 byte proofs with one trusted setup for all fan-in two circuits of $\mathbf{n}$ gates.

Prover does $11\mathbf{n}$ $\mathbf{G}_1$ exp (or $9\mathbf{n}$ $\mathbf{G}_1$ exp with 700 byte proof).

For batch of proofs on same circuit only $3\mathbf{n}$ $\mathbf{G}_1$ exp and 240 bytes for each additional proof.

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

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

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

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

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

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

$\mathsf{open}(\mathbf{f}, \mathbf{i}) := [\mathbf{h}(x)]$, 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 in the Algebraic Group Model.*