Zero knowledge proof from Interactive proof

Sumcheck protocol

$$H = \sum_{b_1,\ldots,b_k \in \{0,1\}} f(b_1, \ldots, b_k)$$

$f(x_1, \ldots, x_k)$

$$f_1(x_1) = \sum_{b_2,\ldots,b_k \in \{0,1\}} f(x_1, b_2 \ldots, b_k)$$

r_1

$$f_2(x_2) = \sum_{b_3,\ldots,b_k \in \{0,1\}} f(r_1, x_2, b_3 \ldots, b_k)$$

....

r_i

$$f_{i+1}(x_{i+1}) = \sum_{b_{i+2},\ldots,b_k \in \{0,1\}} f(r_1, \ldots, r_i, x_{i+1}, b_{i+2} \ldots, b_k)$$

....

r_{k-1}

$$f_k(x_k) = f(r_1, \ldots, r_{k-1}, x_k)$$

Check:

$$H = f_1(0) + f_1(1)$$

$$f_1(r_1) = f_2(0) + f_2(1)$$

....

$$f_i(r_i) = f_{i+1}(0) + f_{i+1}(1)$$

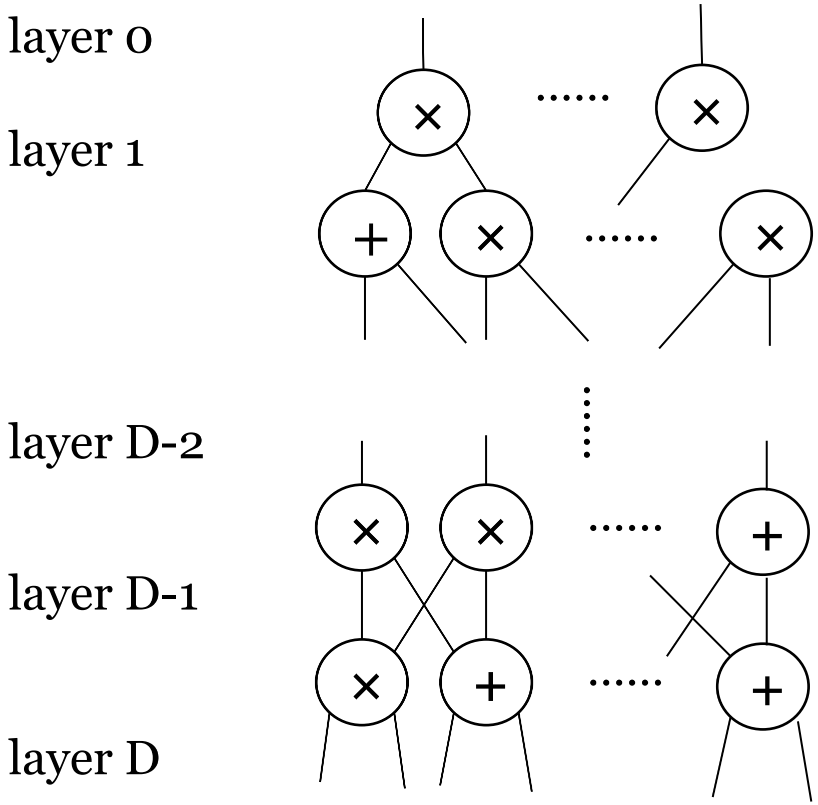
....

$$f_{k-1}(r_{k-1}) = f_k(0) + f_k(1)$$

$$f_k(r_k) = f(r_1, \ldots r_k)$$

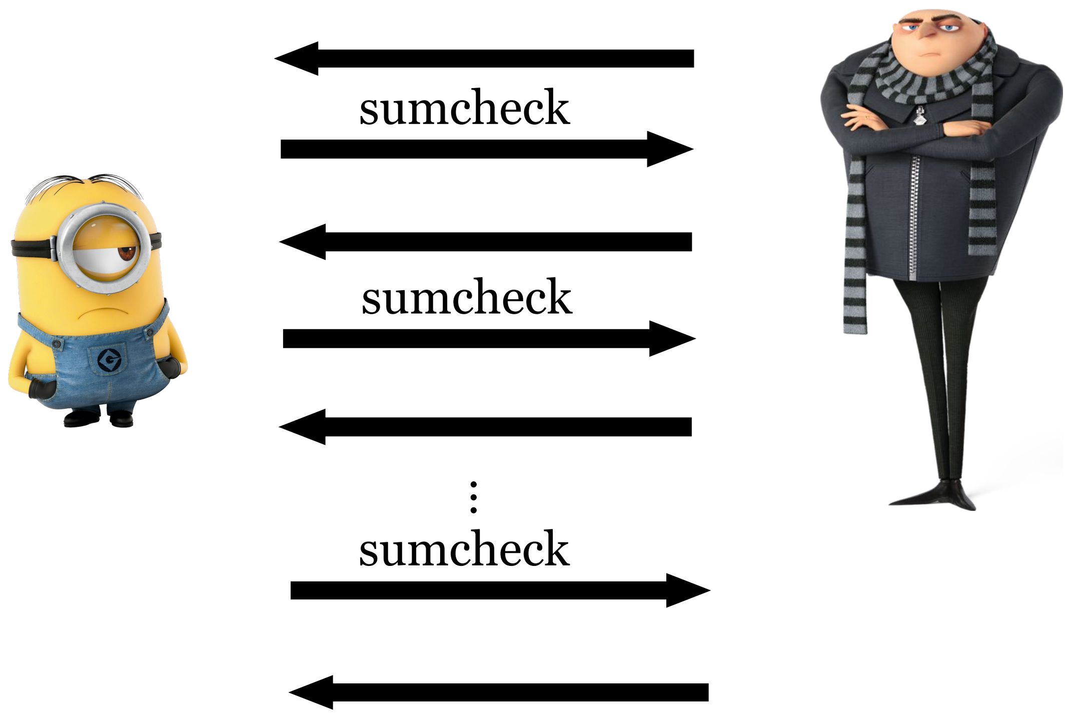
GKR protocol

Output (result)



Input (data)

	output
$V_0(\cdot)$	$V_0(\vec{r_0})$
↓	↓
$V_1(\cdot)$	$V_1(\vec{r_1})$
↓	↓
$V_2(\cdot)$	$V_2(\vec{r_2})$
⋮	⋮
$V_{D-2}(\cdot)$	
↓	
$V_{D-1}(\cdot)$	
↓	↓
$V_D(\cdot)$	$V_D(\vec{r_D})$
	↕
	$V_D(\vec{r_D})$

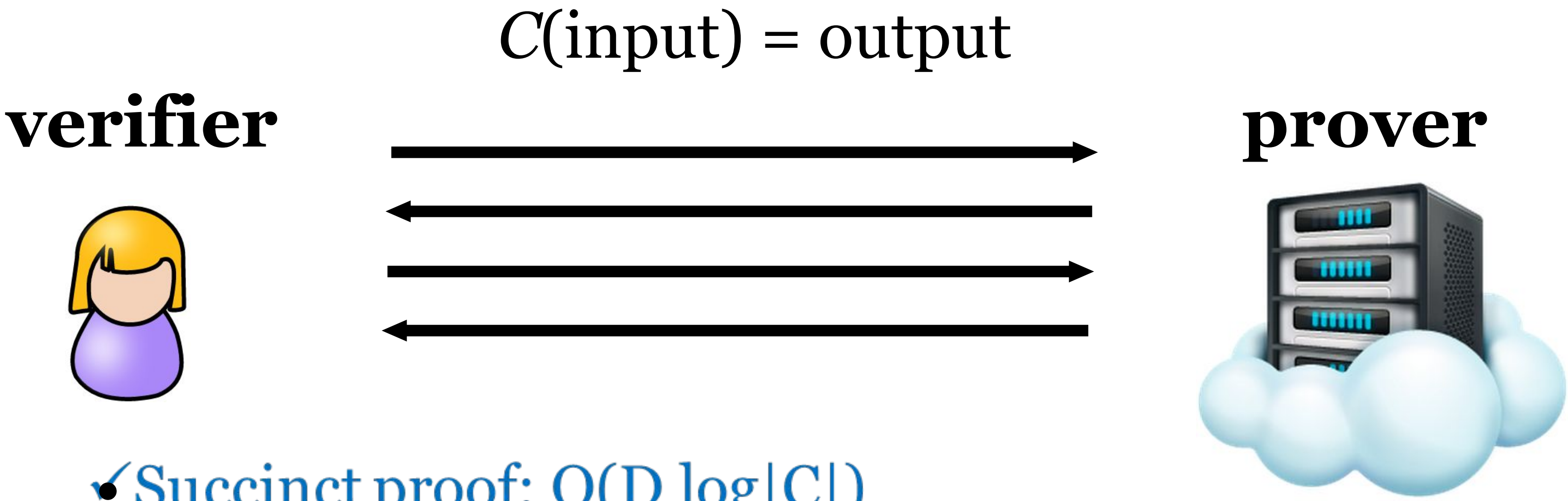


$$V_i(\vec{g}) = \sum_{\vec{u},\vec{v}\in\{0,1\}^s} f_{i,\vec{g}}(\vec{u},\vec{v})$$

$$= \sum_{\vec{u},\vec{v}\in\{0,1\}^s} (add_i\,(\vec{g},\vec{u},\vec{v})(V_{i+1}(\vec{u}) + V_{i+1}(\vec{v}))$$

$$+ \; mult_i(\vec{g},\vec{u},\vec{v})V_{i+1}(\vec{u})V_{i+1}(\vec{v}))$$

Properties of GKR Protocol



✓Succinct proof: O(D log|C|)

✓Succinct verification for structured circuits: O(D log|C|+|x|)

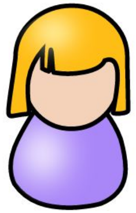
✓Fast prover time: O(|C|) modular add and mult

✓No setup

× Not a proof/argument: verifier computes polynomial $V_D(\overrightarrow{r_D})$ defined by input

Polynomial commitment [KZG10, PST13]

verifier



prover

commitment δ_f

evaluation point a

Verification: ✔ **or** ☐

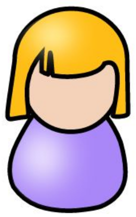
$f(a)$ + proof

$f(x)$

Argument System from GKR [**Z**GK+17]

$C(\text{witness}) = \text{output}$

verifier



prover

witness

Polynomial commitment of $V_D(\cdot)$ defined by witness

GKR

$V_D(\overrightarrow{r_D})$

$\overrightarrow{r_D}$

$V_D(\overrightarrow{r_D})$ and proof

Verification: ✔ **or** ☐

Requirements on complexity

- pk,sk $\leftarrow$Keygen(1^{λ}, d, k)

- δ_f $\leftarrow$ Commit(f, pk)

- v, w $\leftarrow$ Compute(f, pk, a)

- {accept, reject} $\leftarrow$ Verify(pk, δ_f , a, v, w)

Requirements on complexity

- pk,sk $\leftarrow$Keygen(1^{λ}, d, k)
- δ_f $\leftarrow$ Commit(f, pk): constant size
- v, w $\leftarrow$ Compute(f, pk, a)
- {accept, reject} $\leftarrow$ Verify(pk, δ_f , a, v, w)

Requirements on complexity

- pk,sk $\leftarrow$Keygen(1^{λ}, d, k)
- δ_f $\leftarrow$ Commit(f, pk): constant size
- v, w $\leftarrow$ Compute(f, pk, a): logarithmic proof size
- {accept, reject} $\leftarrow$ Verify(pk, δ_f , a, v, w)

Requirements on complexity

- pk,sk $\leftarrow$Keygen(1^{λ}, d, k)
- δ_f $\leftarrow$ Commit(f, pk): constant size
- v, w $\leftarrow$ Compute(f, pk, a): logarithmic proof size
- {accept, reject} $\leftarrow$ Verify(pk, δ_f , a, v, w): logarithmic verification time

Univariate polynomial commitment

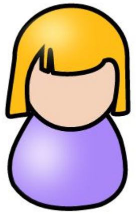
public key: $g, g^{s}, g^{s^2}, g^{s^3}, \ldots, g^{s^d}$

commitment $\delta_f : g^{f(s)} = g^{\sum c_i s^i}$

verifier

prover

evaluation point a



$v = f(a)$ + proof: $w = g^{q(s)}$

$f(x)$

Verification:

$e(\delta_f / g^{f(a)}, g) = e(g^{s-a}, w)$

$f(x) - f(a) = (x - a)q(x)$

Proof

- q-strong Bilinear Diffie-Hellman assumption

given $p, g, g^s, g^{s^2}, g^{s^3}, g^{s^4}, \ldots, g^{s^q}$, cannot compute c, h

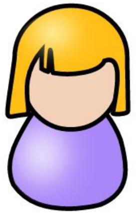
$\text{s.t.} \quad h = e(g,g)^{\frac{1}{s+c}}$

Complexity

public key: $g, g^{s}, g^{s^2}, g^{s^3}, \dots, g^{s^d}$

commitment $\delta_f : g^{f(s)} = g^{\sum c_i s^i}$

verifier



prover

evaluation point a

$v = f(a)$ + proof: $w = g^{q(s)}$

$f(x)$

Verification:

$$e(\delta_f / g^{f(a)}, g) = e(g^{s-a}, w)$$

$$f(x) - f(a) = (x-a)q(x)$$

Commitment: O(1) size and O(d) time

Prover time: O(d)

Proof size: O(1)

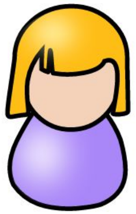
Verification: O(1)

Multivariate polynomial commitment

public key: $g, g^{s_1}, g^{s_2}, g^{s_1 s_2}, \ldots, g^{s_1 s_2 s_3 \cdots s_k}$

commitment $\delta_f : g^{f(s_1, s_2, \ldots, s_k)}$

verifier



evaluation point $(a_1, a_2, \ldots, a_k)$

$v = f(\vec{a})$ + proof: $w_i = g^{q_i(s)}$

prover

$f(\vec{x})$

Commitment: O(1) size and O(n) time

Prover time: O(n)

Proof size: O(k)=O(logn)

Verification: O(k)=O(logn)

Verification:

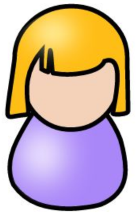
$$e\left(\delta_f / g^{f(\vec{a})}, g\right) = \Pi_{i=1}^{k} e(g^{s_i - a_i}, w_i)$$

$$f(\vec{x}) - f(\vec{a}) = \sum^{k} (x_i - a_i) q_i(\vec{x})$$

Argument System from GKR

$C(\text{witness}) = \text{output}$

verifier



prover

witness

Polynomial commitment of
$V_D(\cdot)$ defined by witness

GKR

$V_D(\overrightarrow{r_D})$

$\overrightarrow{r_D}$

$V_D(\overrightarrow{r_D})$ and proof

Verification: ✔ **or** ☐

Prover time: O(|C|)
Proof size: O(D log|C|)
Verification time: O(D log|C|)

Follow-up work [XZZS20]

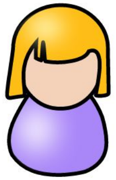
- Polynomial commitment without trusted setup
  - Prover time: O(n log n)
  - Proof size: $O(\log^2 n)$
  - Verification time: $O(\log^2 n)$

Open problem: Polynomial commitment with linear prover time even with $O(\sqrt{n})$ proof size and linear verification time

Leakage of sumcheck/GKR Proof

$$V_i(\vec{g}) = \sum_{u,v \in \{0,1\}^{\log|C|}} mult_i(\vec{g},\vec{u},\vec{v})V_{i+1}(\vec{u})V_{i+1}(\vec{v}) + add_i\,(\vec{g},\vec{u},\vec{v})(V_{i+1}(\vec{u}) + V_{i+1}(\vec{v}))$$

verifier **prover**



$f_1(x_1) = \sum_{b_2,\dots,b_{\log n} \in \{0,1\}} f(x_1, \dots, b_{\log n})$

r_1

$f_2(x_2) = \sum_{b_3,\dots,b_{\log n} \in \{0,1\}} f(r_1, x_2, \dots, b_{\log n})$

r_2

$f_{\log n}(x_{\log n}) = f(r_1, r_2, \dots, x_{\log n})$

Weighted sums of values in the circuit

Making GKR zero knowledge [XZZPS, crypto19]

Masking polynomial

$$H + r\Delta = \sum_{b_1,\ldots,b_{\log n}\in\{0,1\}} \left(f(b_1,\ldots,b_{\log n}) + r\delta(b_1,\ldots,b_{\log n})\right)$$

- mask with **small** random polynomials
- size of $\delta(\ )$ is only **O(log $|C|$)**: same size/entropy as the proof
- $\delta(x_1,\ldots,x_{\log n}) = \delta_1(x_1) + \delta_2(x_2) + \cdots + \delta_{\log n}(x_{\log n})$
- almost no overhead in practice

Comparison to SNARK

	SNARK	GKR-based
Setup	O(C) trusted setup	none
Prover time	O(C logC), exponentiations	O(C+n logn), add and mult
Proof size	O(1), 200 Bytes	$O(D\log C+\log^2 n)$, 200KB
Verification time	O(1), 3ms	$O(D\log C+\log^2 n)$, 40ms