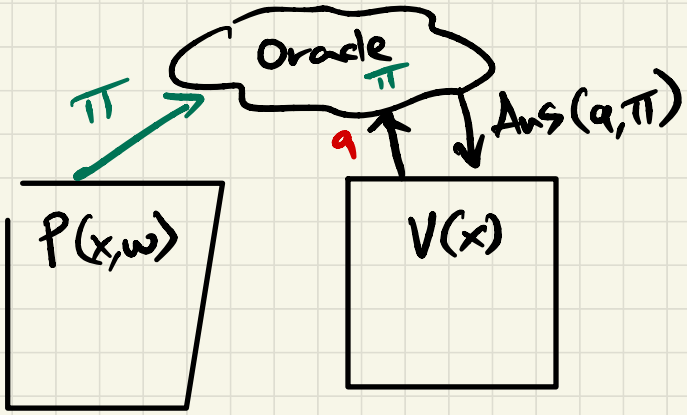
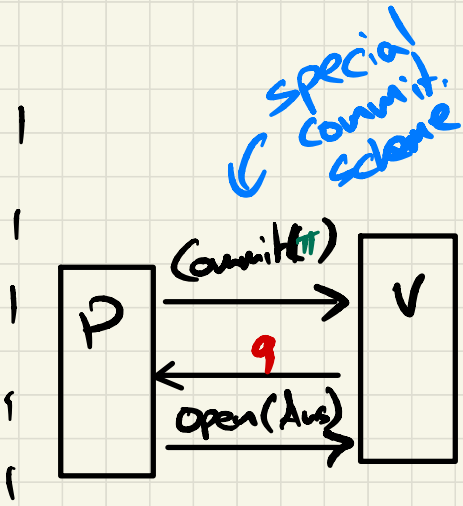
Privacy Enhancing Technologies

4. SNARGS

Recap on SNARGs

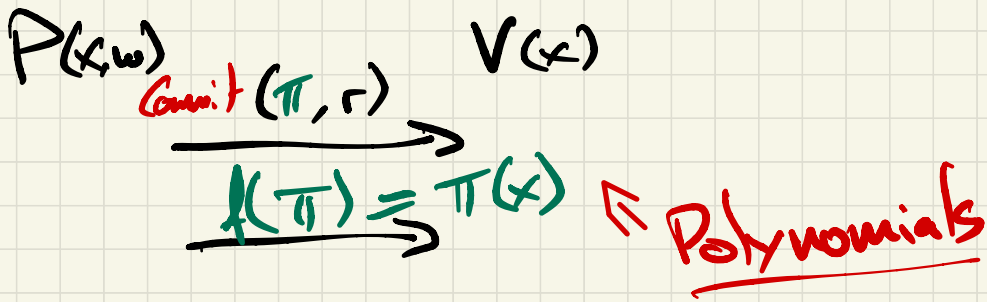




Properties:
- proof is short: $O(\lambda, \text{poly}\log|C|)$
- verif is efficient: $O(\lambda, |x|, \text{poly}\log|C|)$

Terminology

- SNARG = Succinct Non-interactive ARGument
- SNARK = ⊕ Proof-of-knowledge
- zk-SNARK = ⊕ zero knowledge

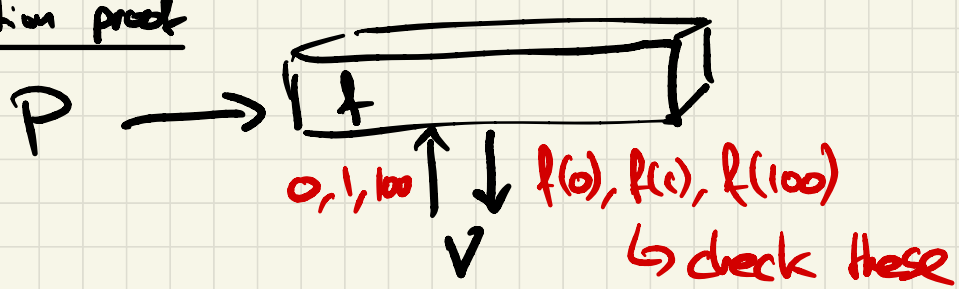


I'm going to prove that Fib(100) = 354...075

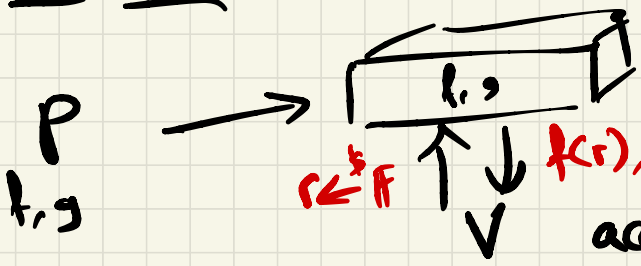
$$f(x) = \underbrace{a_0}_{\in F} + \underbrace{a_1}_{\in F} x + a_2 x^2 + \dots + \underbrace{a_d}_{\in F} x^d$$

1. $f(0) = f(1) = 1$
2. $f(100) = 354\dots075$
3. $f(x+2) = f(x) + f(x+1) \quad \forall x \in \{0\dots$

Evaluation proof



Equality test



accept iff $f(r) = g(r)$

Why is this sound?

> A non-zero poly over $\mathbb{F}$ of degree d has $\leq d$ roots over $\mathbb{F}$
>
> roots: $x \in \mathbb{F}$ s.t. $f(x) = 0$

Suppose prover cheats and $f \neq g$

Then $z = f - g$ is a non-zero poly of degree $\leq d$

So z has at most d roots

$$\mathbb{P}_r[z(r) = 0 \mid f \neq g] \leq \frac{d}{|\mathbb{F}|} = \text{neg}(\lambda)$$

$z(r) = 0$

$\Leftrightarrow f(r) - g(r) = 0$

$\Leftrightarrow f(r) = g(r)$

Zero Test

let's show $f(x+2) - f(x+1) - f(x) = 0$
$\forall x \in \{0, 1, \dots, 98\}$

let's prove that $f(x) = 0 \quad \forall x \in \Omega$

> f is zero on Ω iff $f(x)$ is divisible by $z_\Omega(x) = \prod_{a \in \Omega} (x - a)$

Ex: $f(x) = x^3 - 7x + 6 \qquad \Omega = \{1, 2\}$

$f(1) = f(2) = 0$

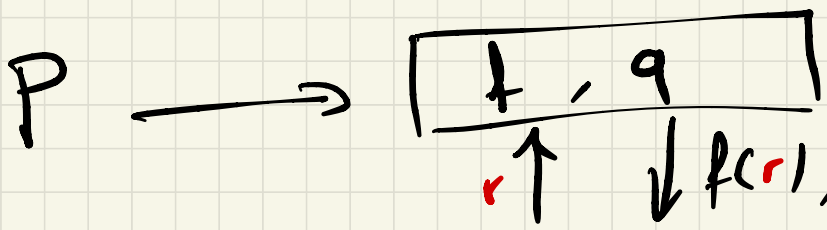
$f(x) = \underbrace{(x-1)(x-2)}_{z_\Omega(x)} \cdot (x+3)$

Zero test

vanishing poly ← $Z_\Omega(x)$ · $q(x)$ → quotient poly

Write $f(x) = Z_\Omega(x) \cdot q(x)$

f: deg $\le d$; Z_Ω: deg $\le |\Omega|$; q: deg $\le d$



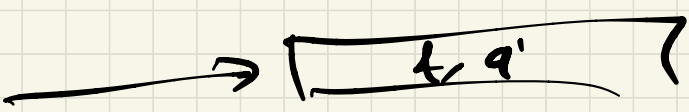
V accept if $f(r) = Z_\Omega(r) \cdot q(r)$

(compute locally)

Why sound?

Assume f is not zero on Ω

Then $f(x) / Z_\Omega(x)$ doesn't exist (as a poly)



This means $f(x) - q'(x) \cdot Z_\Omega(x) \neq 0$

and has degree $\le 2d$

$$\mathbb{P}[f(r) - q(r) \cdot t(r) = 0] \le \frac{2d}{|\mathbb{F}|}$$

The full Fibonacci proof

$f(x+2) - f(x+1) - f(x) = 0 \quad \forall x \in \{0, \ldots, 98\}$

P $q(x) = \dfrac{f(x+2) - f(x+1) - f(x)}{\prod_{a=0}^{98} (x-a)}$

$f(x)$, $q(x)$

$0, 1, 100$

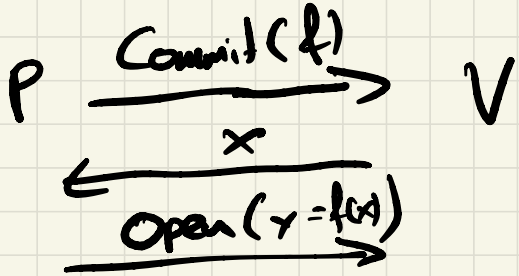
r

$f(0), f(1), f(100)$

$f(r+2) - f(r+1) - f(r)$

$= q(r) \cdot \prod_{a=0}^{98} (r-a)$

Polynomial Commitments



Def

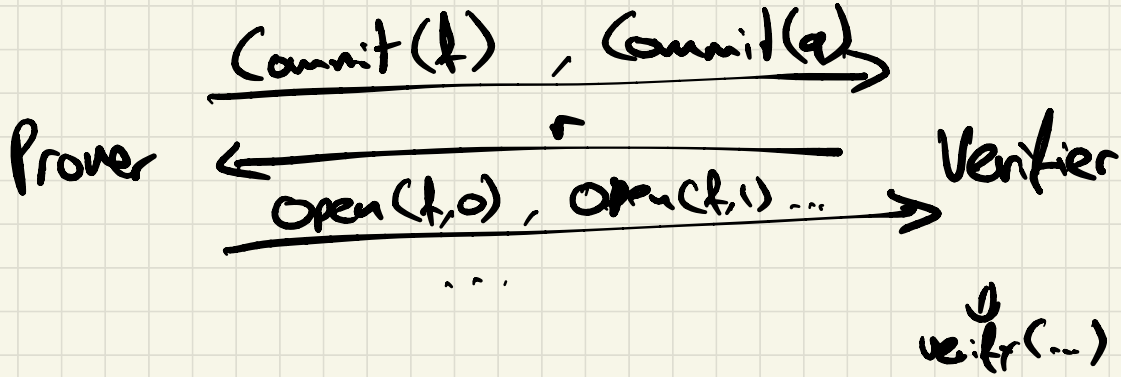
- $\text{Setup}(d, \lambda) \to PP$
- $\text{Commit}(pp, f) \to C$ ("shorter" than f)
- $\text{Open}(pp, f, x) \to \pi$ (proof that $f(x) = y$)
- $\text{Verify}(pp, C, x, y, \pi) \to \{0, 1\}$

Correctness

$$1 = P_r\left[\text{Verify}(pp, C, x, f(x), \pi) = 1 : \begin{array}{l} pp \leftarrow \text{Setup}(d, \lambda) \\ C \leftarrow \text{Commit}(pp, f) \\ \pi \leftarrow \text{open}(pp, f, x) \end{array}\right]$$

"Evaluation Binding"

$$\Pr\left[\begin{array}{l} \mathsf{Verif}(c,x,y,\pi)=1 \\ \mathsf{Verif}(c,x,y',\pi')=1 : \\ y \neq y' \end{array} \quad \begin{array}{r} pp \leftarrow \mathsf{Setup}(1^\lambda) \\ (c,x,y,\pi,y',\pi') \leftarrow A(pp) \end{array}\right] \leq \mathsf{negl}(\lambda)$$



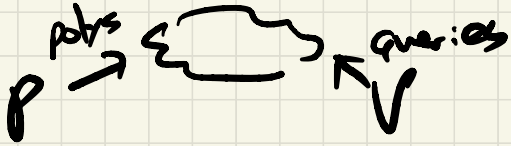
Non-interactivity : Fiat-Shamir

Succinctness: |Commit| , |open| are $O(\lambda, \mathsf{poly}(\log(t)))$

HW2: build a Poly commit scheme (partially)

PLONK

A "polynomial IOP"

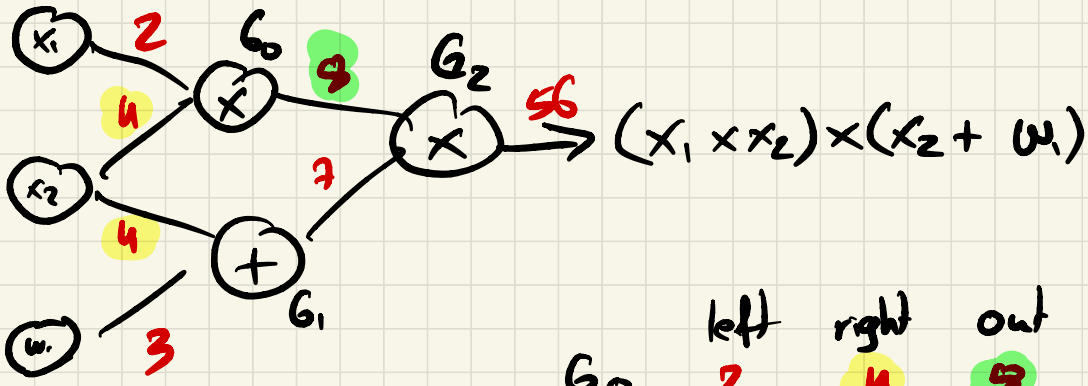


Setting: $C(x, w) = 0$

- x: public input
- w: witness (private)

Goal: build polynomials that encode this statement

Circuit traces



	left	right	out
G_0	2	4	8
G_1	4	3	7
G_2	8	7	56

- $|C|$ = # of gates
- $|I| = |I_x| + |I_w|$
- degree $d = 3 \cdot |C| + |I|$
- $\Omega = \{1, \alpha, \alpha^2, \ldots \alpha^{d-1}\}$

Trace polynomial f_{trace} of deg d

1. encode inputs: $f_{trace}(\alpha^{-j}) =$ input #j
2. encode wires: for $\ell = 0 \ldots |C| - 1$
   $f_{trace}(\alpha^{3\ell}) =$ left input to gate ℓ
   " $(\alpha^{3\ell+1}) =$ right "
   " $(\alpha^{3\ell+2}) =$ output "

the prover commits to f_{trace} ∀ ∘

How to verify?

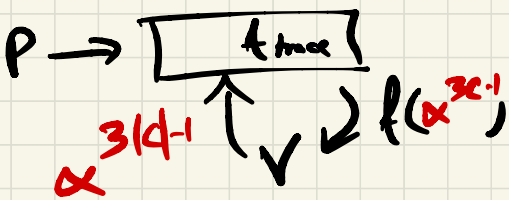
What to verify?

$C(x, w) = 0$

1. the output of the circuit (the last gate) = 0
2. the correct inputs x are used
3. the gates are evaluated correctly
4. the circuit wiring is correct

(3 and 4: see notes)

1) Checking output



2) Checking inputs

- $f_{tr}(\alpha^{-1}) = x_1 \quad \dots \quad f_{tr}(\alpha^{-k}) = x_k$
- Verifier build f_{in} that encodes inputs

  $f_{in}(\alpha^{-j}) = x_j$ for all public inputs

- Prove

$$f_{tr}(x) - f_{in}(x) = 0$$
$$\text{for } x \in \{\alpha^{-1}, \dots, \alpha^{-|I_x|}\}$$

PLONK SNARG

"Universal" Setup: runs Poly-Commit setup to get pp

Circuit-specific setup: "do some work in time $O(|C|)$ that depends only on C"

Proof:
- Prover runs $C(x,w)$ and creates ftrace and commits to it
- Verifier queries ftrace at a bunch of points