

Privacy Enhancing Technologies

SNARKs

Florian Tramer

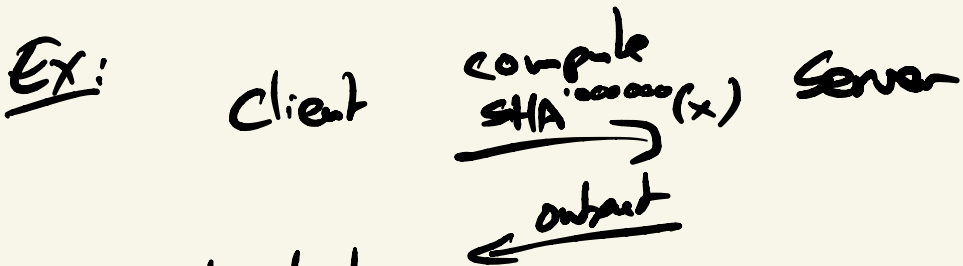
- We saw interactive proofs (+2k)
- And how to make them non-interactive
- Now: we want to make proofs short and efficient

$$P \xrightarrow{\pi} V$$

↳ size of π to be small

↳ running time of V to be small

(↳ " P to be n -linear")



to check,
need to rerun
the computation...

SNARG : Succinct Non-interactive Argument
(SNARK) : + PoK
(zk-SNARK) + zk

↳ "proof"
with non-
perfect
soundness

What does succinct mean?

* Short proofs : $|\pi|$ is poly-log of the
witness and statement

$$|\pi| = O(\lambda, \text{polylog}(|x|, |w|))$$

↳ $(\log(x))^k$

* efficient verification : V runs in
time $O(\lambda, |x|, \text{polylog}(|w|))$

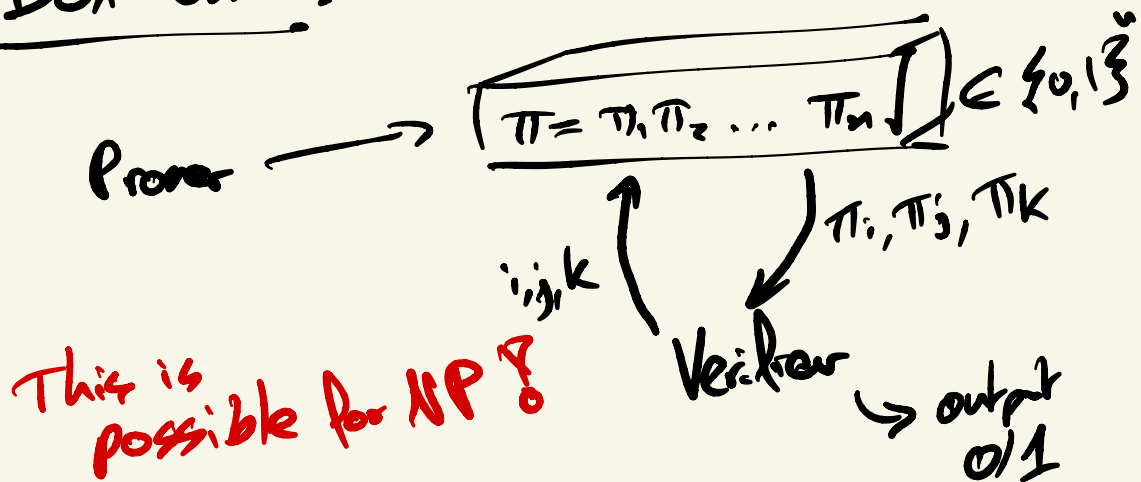
This seems highly non trivial.

Even if we don't care about zk,
NP proofs don't satisfy this

How to build a SNARG:

- 1) you build a proof system with information-theoretic soundness (eg a DP proof) but with some special succinct-like properties and where the prover and verifier interact in a "special" way
- 2) Use cryptography to force P^* to act in this "special" way

Box Game:



PCP Theorem: For any NP language,
the prover can create a proof π of
length $\text{poly}(|w|, |x|)$ s.t the verifier
reads 3 random bits and accepts
with:

- perfect completeness
- soundness error $\leq 1/2$

"succinctness": the verifier only reads
 $O(\log n)$ random bits
and receives $O(1)$ bits
in return

From PCP to SNARGs:

* The PCP proof system is information
theoretic

* Use cryptography to "instantiate" this
protocol in a succinct way

The ingredients:

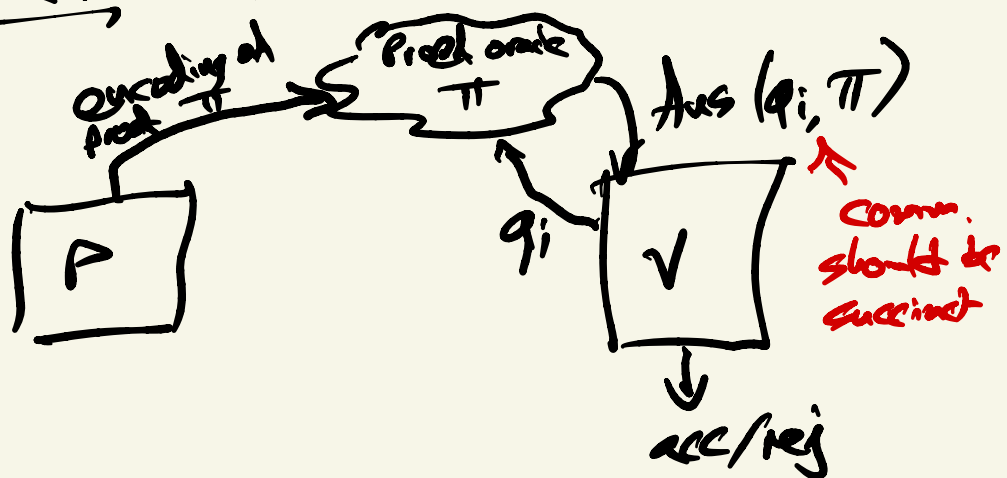
- Prover commits to Π
 - ↳ such that $|\text{commit}| = O(\text{poly} \log |\Pi|)$
- Verifier can request an opening at any index
 - verifier sends i
 - prover sends x_i, Π_i to prove that c commits to a string x with $x_i = \gamma$ with $|\Pi_i| = O(\text{poly} \log |x|, |c|)$

⇒ this is called a vector commitment

↳ e.g. Merkle Tree

⇒ "CS Proofs" Nicali

Abstraction: Interactive Oracle Proof



this is information theoretic. should be

- complete
- sound

Types of IOPs:

- PCPs (intractable)

- linear IOPs: proof is encoded as a vector $\Pi \in \mathbb{F}^n$ and the verifier queries are vectors $q \in \mathbb{F}^n$ and learns $\langle \Pi, q \rangle \in \mathbb{F}$

- polynomial IOP: the proof is encoded as a polynomial $\Pi(x)$ over $\mathbb{F}$ and the verifier asks evaluation queries $\Pi(z)$ for $z \in \mathbb{F}$
-

Why polynomials? they're cool!

idea: a polynomial is a "succinct" way of representing many relations or equations

Mini example:

Statement: the 100-th Fibonacci number is 354, ..., 075

to convince you, I create $f_{100}(x)$ over $\mathbb{F}_p$ and claim:

$$1. f(0) = f(1) = 1 \quad (\text{initial Fib numbers})$$

$$2. f(100) = 354, \dots, 075$$

$$3. f(x+2) = f(x) + f(x+1) \quad \forall x \in \{0, 1, \dots\}$$

(Fib recurrence)

Assume we have a "polynomial Box"

$$P \rightarrow \boxed{f(x)}$$

$x \uparrow \quad \downarrow f(z)$
V

Proving 1) and 2) is immediate

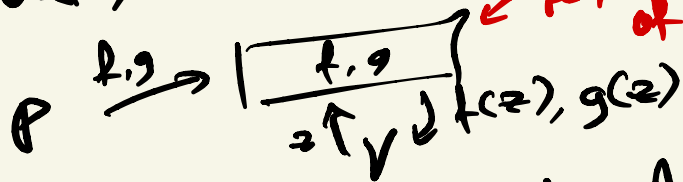
$$P \rightarrow \boxed{f_{fib}(x)}$$

$0, 1, 100 \uparrow \quad \downarrow f(0), f(1), f(100)$
V and checks them

Proving 3) is where it gets fun!

Proving global properties of polynomials

1) Equality test



← polynomials in $\mathbb{F}[x]$
of degree $\leq d$

decide if $f(x) = g(x)$

V queries a random $r \xleftarrow{\$} \mathbb{F}$
and accepts iff $f(r) = g(r)$

- This is complete
- The soundness error is $\frac{d}{|\mathbb{F}|}$

Proof:

Thm (Fundamental Thm of algebra):

A non-zero polynomial over $\mathbb{F}$ of degree d has at most d roots over $\mathbb{F}$

Suppose $f \neq g$. Then $f(x) - g(x)$
is a ^{non-zero} polynomial of degree $\leq d$

So $f - g$ has at most d roots!
the probability that r is one of them
is $\leq \frac{d}{|\mathbb{F}|}$.

Lemma: Schwartz-Zippel lemma

$$f(x, y, z) = 3x^2 \overset{\text{deg } 6}{y^3} z + 2xy^2 \overset{\text{deg } 3}{} z$$

has total degree 6

- let $f(x_1, \dots, x_n) \in \mathbb{F}[x_1, \dots, x_n]$
be a non-zero poly of total degree d
- let Ω be a (finite) subset of $\mathbb{F}$ and
let $r_1, \dots, r_n \stackrel{i.i.d.}{\leftarrow} \Omega$

Then $\Pr[f(r_1, \dots, r_n) = 0] \leq \frac{d}{|\Omega|}$

An extension of equality test: zero test

Prove that some polynomial is zero over a set Ω

How we'll use this:

Prove that $f(x+2) - f(x) - f(x+1) = 0$
for $x \in \{91, \dots, 98\}$

Def: The vanishing polynomial over Ω
is $z_\Omega(x) = \prod_{a \in \Omega} (x - a)$

Thm: f is zero on Ω iff
 $f(x)$ is divisible by $z_\Omega(x)$
over $\mathbb{F}$
($f(x) = z_\Omega(x) \cdot q(x)$)
 $\deg q \leq d$

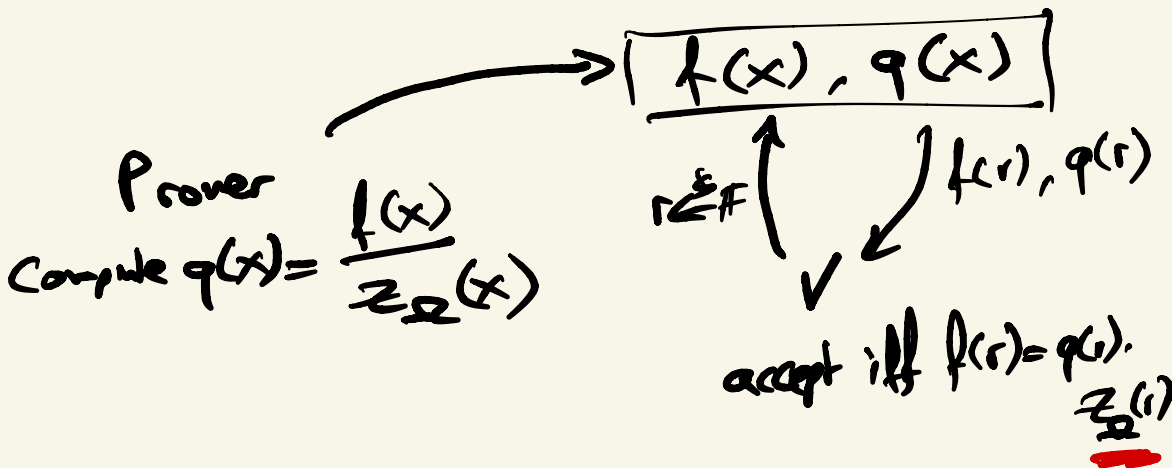
Example: $f(x) = x^3 - 7x + 6$

$$\Omega = \{1, 2\}$$

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

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

The Poly-IOP:



Lemma: the Zcash protocol has

- perfect completeness
- soundness error $2d/|\mathbb{F}|$

Proof: Assume f is not zero on Ω

$$\text{So } f(x) \neq z_\Omega(x) \cdot q(x) \\ \text{for any } q(x) \text{ of } \deg \leq d$$

So the prover has to commit to
some polynomial $q(x)$ of $\deg \leq d$

$$\text{Look at } f(x) - q(x) \cdot z_\Omega(x)$$

This is a non-zero poly of
degree $\leq 2d$

$$\text{Then, } \mathbb{P}[f(r) - q(r) \cdot z_\Omega(r) = 0] \\ \leq \frac{\deg}{|\mathbb{F}|} = \frac{2d}{|\mathbb{F}|}$$

$$q(x) = \frac{f(x+2) - f(x+1) - f(x)}{\prod_{i=0}^{38} (x-i)} \quad \boxed{f(x), q(x)}$$

Prover

$0, \dots, 100$
 $r, r+1, r+2$

$f(0), f(1), f(100)$
 $f(r), f(r+1), f(r+2)$
 $q(r)$

check that

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

$$\underline{\underline{Z_{\Omega}(r)}}$$

Sandwich error $\approx \frac{200}{|H|}$

⚠ computing $Z_{\Omega}(r)$ is not succinct...

Solution: In the SNARK we'll look at (PLONK), Ω will always have some "special structure" so that we can compute $Z_{\Omega}(1)$ in time $\text{polylog}(|\Omega|)$

Road to a SNARK:

- * A Poly-IO for all of NP: PLONK
- * A polynomial commitment scheme!
 - ↳ many options, different assumptions
 - trust models
 - efficiency

Polynomial Commitment Schemes

- $\text{Setup}(d, \lambda) \Rightarrow \text{pp}$ (public parameters)
 $d \leftarrow$ a bound on the degree
- $\text{Commit}(\text{pp}, f) \Rightarrow \text{commitment } c$
to $f \in \mathbb{F}^{\text{sd}}[x]$
- $\text{Open}(\text{pp}, c, f, x) \Rightarrow \pi$ proves that
 $y = f(x)$
- $\text{Verify}(\text{pp}, c, x, y, \pi) \Rightarrow \{\text{acc}, \text{rej}\}$

Properties:

- Correctness: $\forall d \in \mathbb{N}, \forall f \in \mathbb{F}^{\text{sd}}[x]$
 $\forall x \in \mathbb{F}$

$$P_c \left[\text{Verify}(\text{pp}, c, x, f(x), \pi) = 1 \mid \begin{array}{l} \text{pp} \leftarrow \text{Setup}(d, \lambda) \\ c \leftarrow \text{Commit}(\text{pp}, f) \\ \pi \leftarrow \text{Open}(\text{pp}, c, x) \end{array} \right] = 1$$

• Evaluation Binding: $\forall d \in \mathbb{N}$ and poly-time Adv A

$$Pr \left[\begin{array}{c} \text{verify}(pp, c, x, y, \pi) = 1 \\ \text{verify}(pp, c, x', y', \pi) = 1 \\ y \neq y' \end{array} \middle| (c, x, y, \pi, y', \pi') \leftarrow A(pp) \right] \leq \text{neg}(1)$$

Our Fiat SNARG:

