ZNARKs

SNARKs for Integer Computations

Matteo Campanelli*
(Offchain Labs)

&

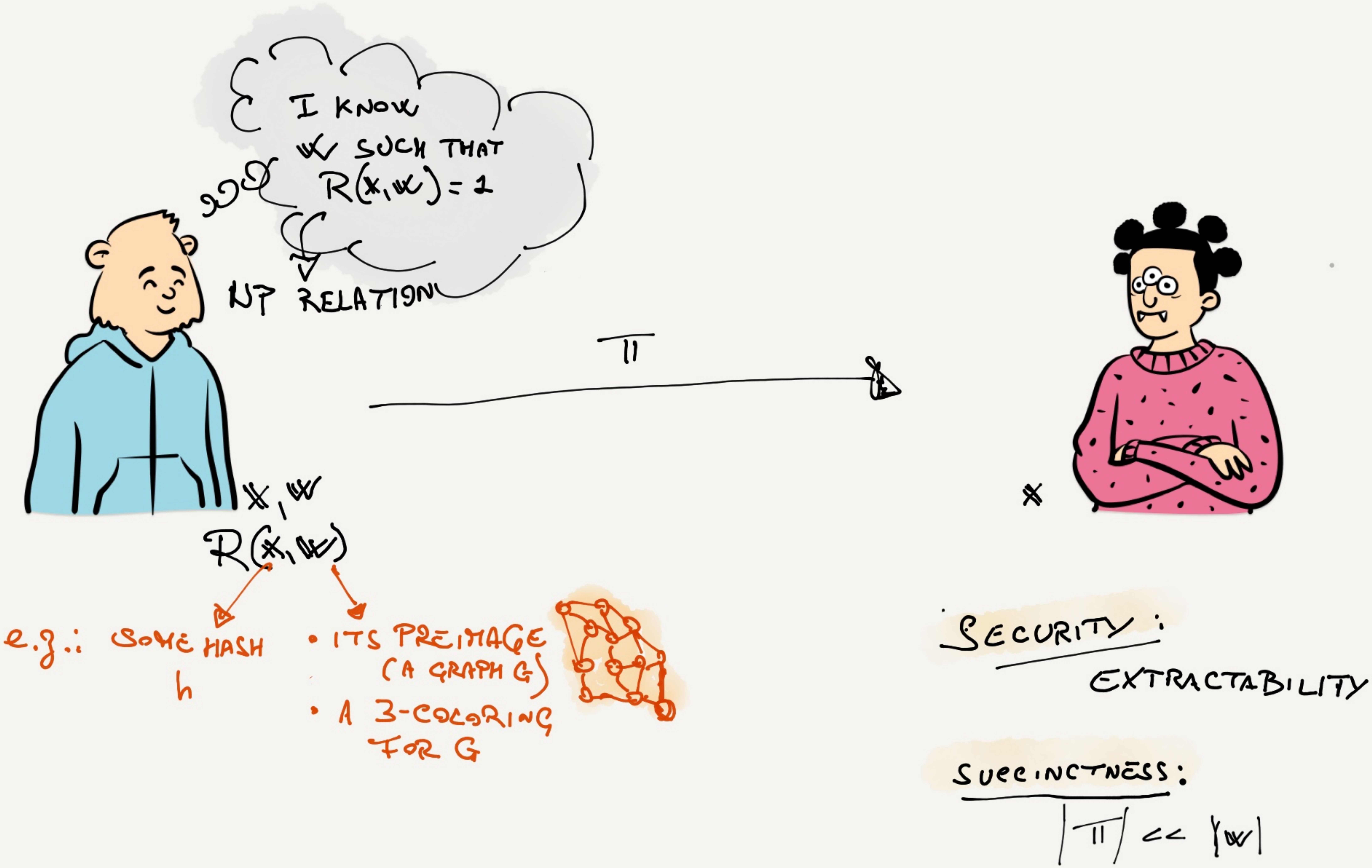
Mathias Hall-Andersen**
(zkSecurity)

"Fully Succinct Arguments over the Integers from First Principles"

ePrint: 2024/1548

* binarywhales.com
** rot256.dev

SNARK BRUSHUP



THE PRESENT: EFFICIENT SNARKS FOR MANY / COMPUTATIONS

CONCRETELY

- A CLASSICAL EXAMPLE:

GROTH16

THE PRESENT: EFFICIENT SNARKS FOR MANY COMPUTATIONS

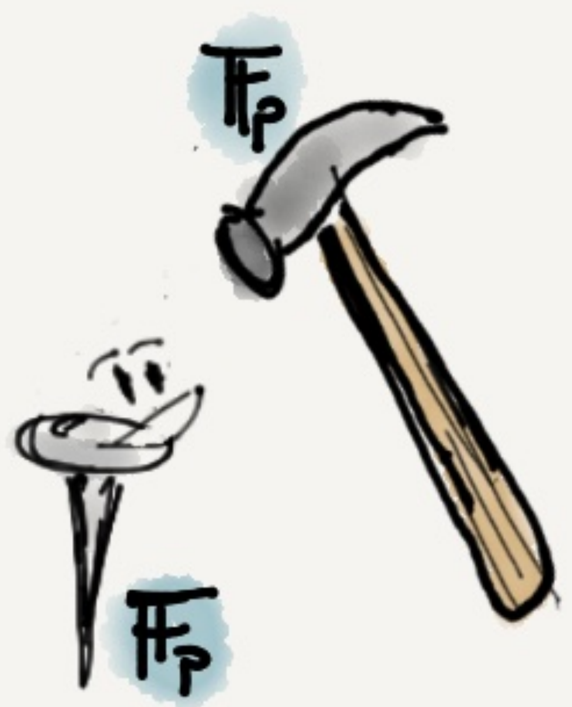
CONCRETELY

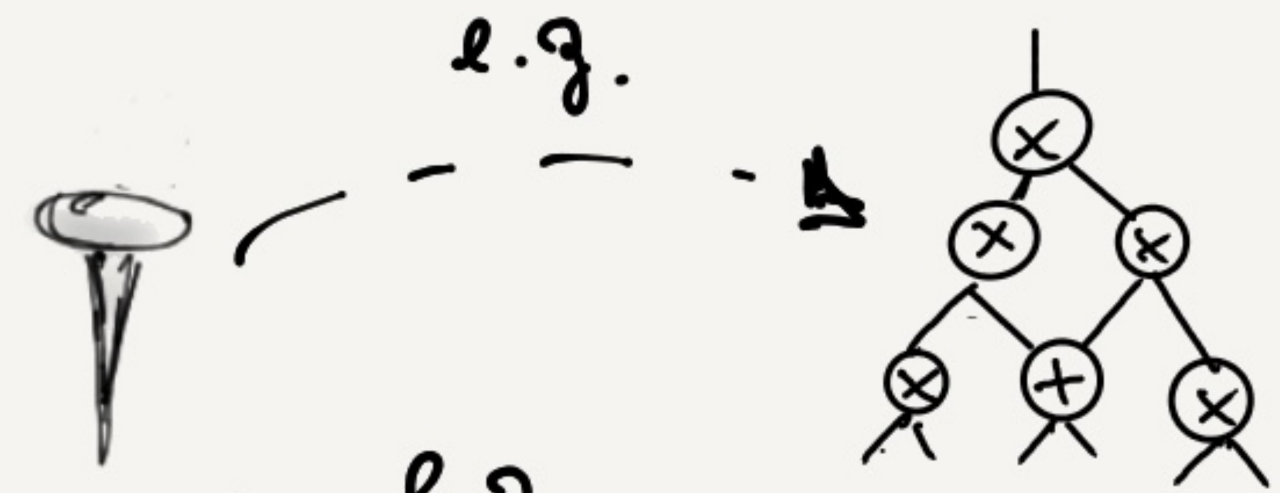
- A CLASSICAL EXAMPLE: $\boxed{\text{GROTH16}}$
- BUT THAT IS NOT THE END OF THE LINE.

VIRGO LIGERO DARK
VAMPIR HYRAX DEW
SPARTAN LUNAR
BASILISK ECLIPSE WHIR/STEER/FRI*
BULLETPROOF *PLONK*
BASEFOLD ORION
NOVA JOLT

(AND MORE...)

CAVEAT: GOOD EFFICIENCY RELIES ON "GOOD REPRESENTATION"

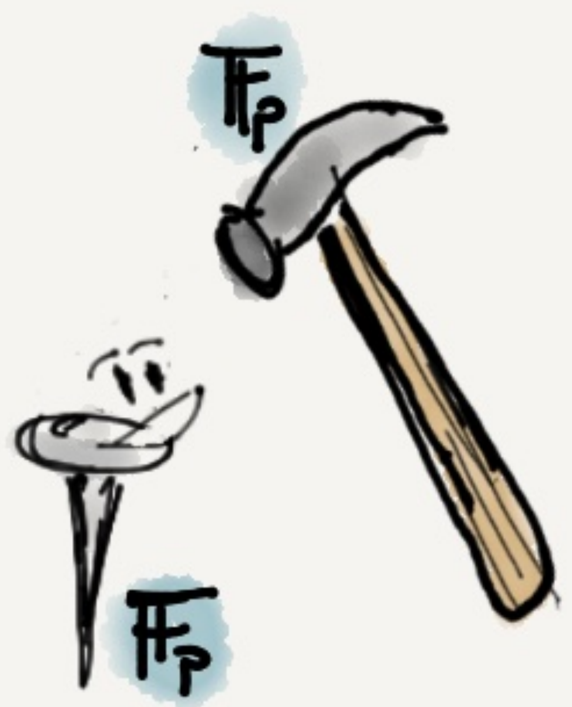


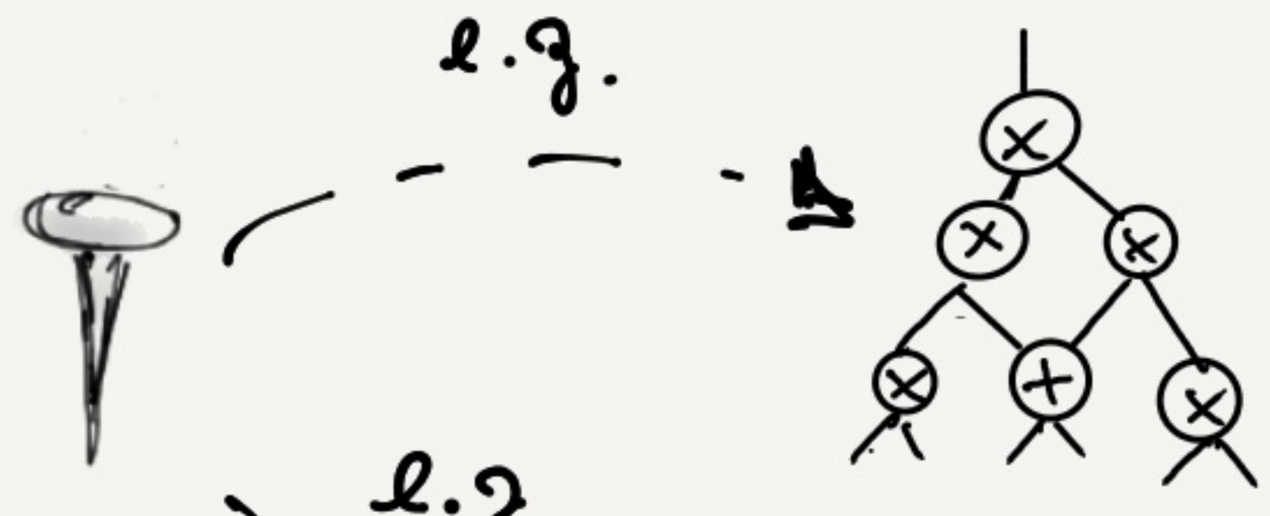


e.g.

$$(A \cdot z) \circ (B \cdot z) - (C \cdot z) = \bar{0}$$

CAVEAT: GOOD EFFICIENCY RELIES ON "GOOD REPRESENTATION"





$$\left(\boxed{A}\cdot z\right)\circ\left(\boxed{B}\cdot z\right)-\left(\boxed{C}\cdot z\right)=\bar{0}$$

BEYOND FINITE FIELDS?

- NOT ALL COMPUTATIONS ADMIT "EASY" REPRESENTATIONS OVER FINITE FIELDS.

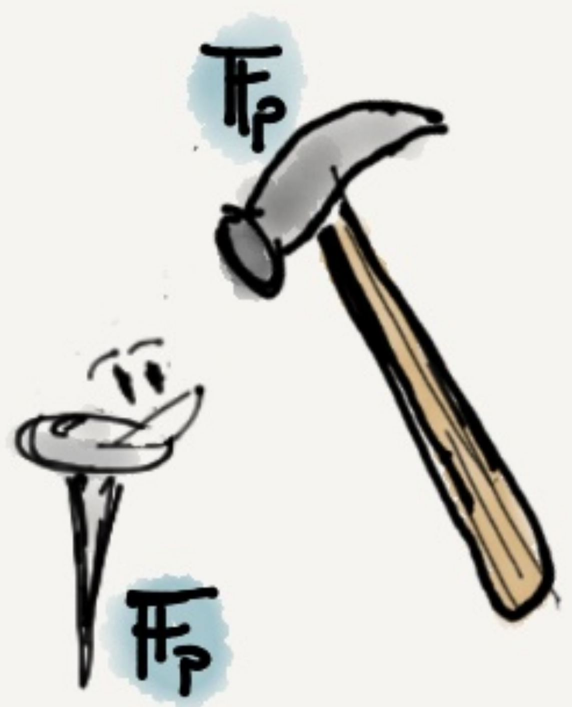
EASY EXAMPLE: RSA SIGNATURE VERIFICATION

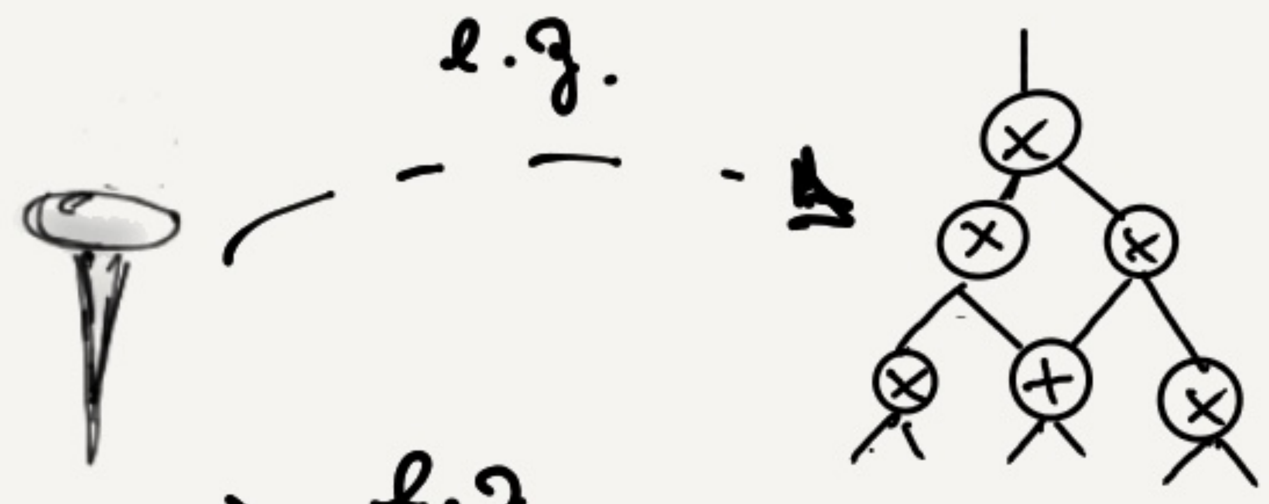
$$\mathrm{Verify}(pk, m, \sigma) := \sigma^e \equiv m \pmod{N}$$

BUT ALSO:

- RSA ACCUMULATORS / ENCRYPTION.
- CRYPTOGRAPHY OVER RINGS (e.g. FHE)
- SCIENTIFIC AND NUMBER-THEORETIC STATEMENTS
- ML

CAVEAT: GOOD EFFICIENCY RELIES ON "GOOD REPRESENTATION"





$$(A \cdot z) \circ (B \cdot z) - (C \cdot z) = 0$$

BEYOND FINITE FIELDS?

- NOT ALL COMPUTATIONS ADMIT "EASY" REPRESENTATIONS OVER FINITE FIELDS.

EASY EXAMPLE: RSA SIGNATURE VERIFICATION

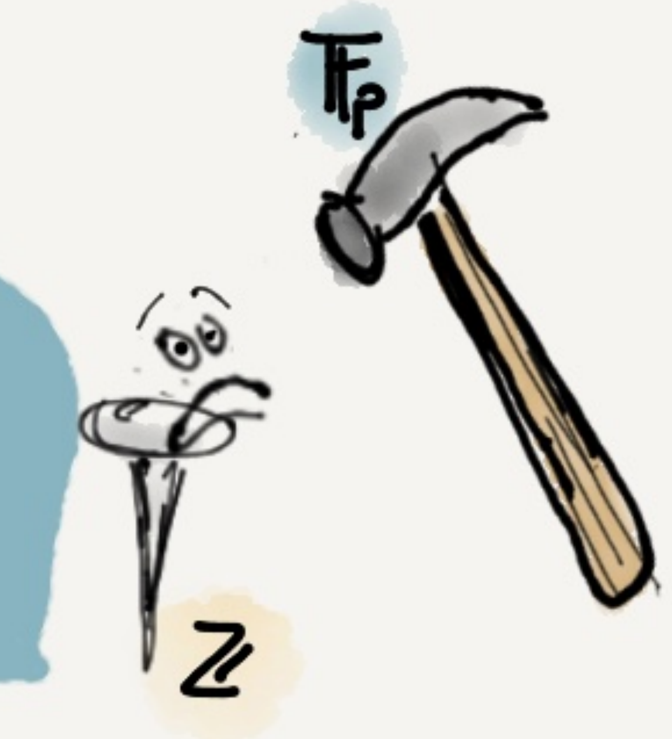
$$\text{Verify}(pk, m, \sigma) := \sigma^e \equiv m \pmod N$$

BUT ALSO:

- RSA ACCUMULATORS / ENCRYPTION.
- CRYPTOGRAPHY OVER RINGS (e.g. FHE)
- SCIENTIFIC AND NUMBER-THEORETIC STATEMENTS
- ML

COMMON THREAD: THE INTEGERS

THE COST OF EMULATING $\mathbb{Z}$ COMPUTATIONS



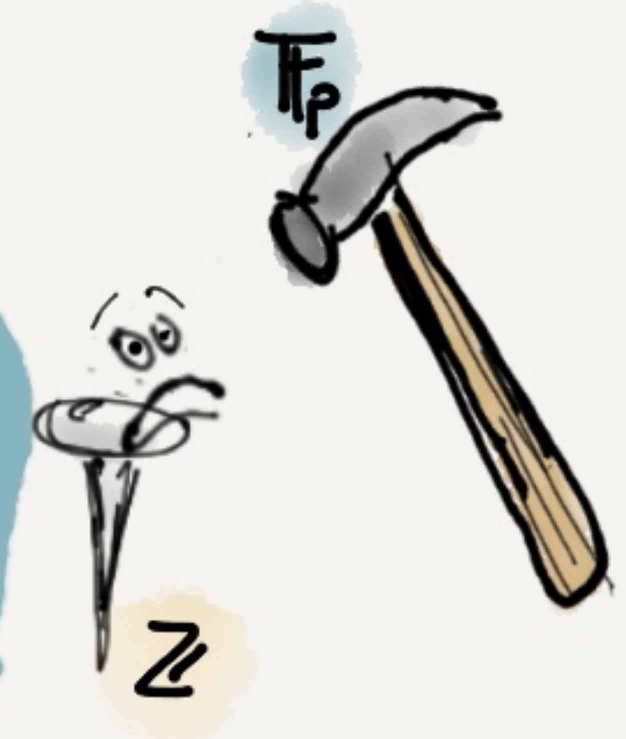
$\boxed{\text{CASE STUDY}}$: RSA SIGNATURES

$$\sigma^e \equiv m \pmod{N}$$

SIGNATURE MESSAGE PK

e IS TYPICALLY 65537 $(= 2^{16}+1)$

THE COST OF EMULATING $\mathbb{Z}$ COMPUTATIONS



$\boxed{\text{CASE STUDY}}$: RSA SIGNATURES

$$\sigma^e \equiv m \pmod N$$

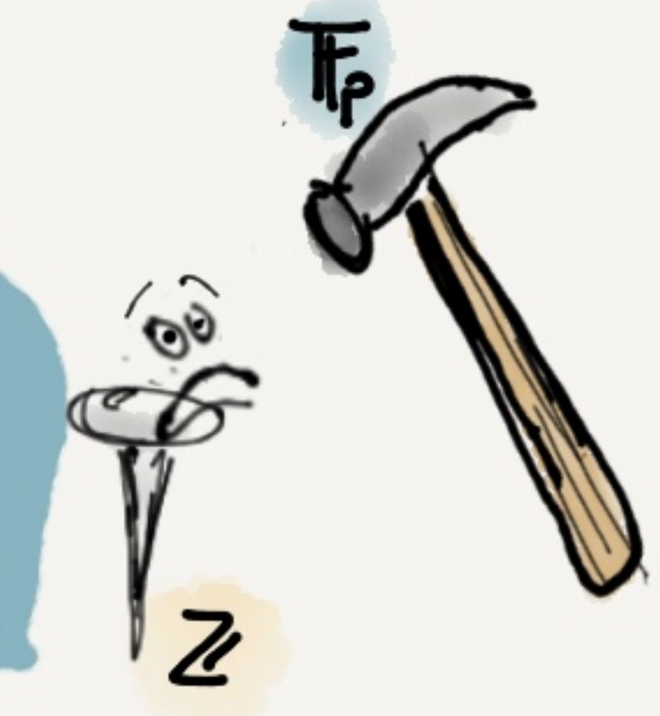
SIGNATURE MESSAGE PK

e is TYPICALLY 65537 $(= 2^{16}+1)$

EQUIVALENT TO: "I know $q \in \mathbb{Z}$:

$$qN + m = \sigma^e \text{"}$$

THE COST OF EMULATING $\mathbb{Z}$ COMPUTATIONS



CASE STUDY: RSA SIGNATURES

$$\sigma^e \equiv m \pmod{N}$$

SIGNATURE MESSAGE PK

e is TYPICALLY 65537 $(= 2^{16}+1)$

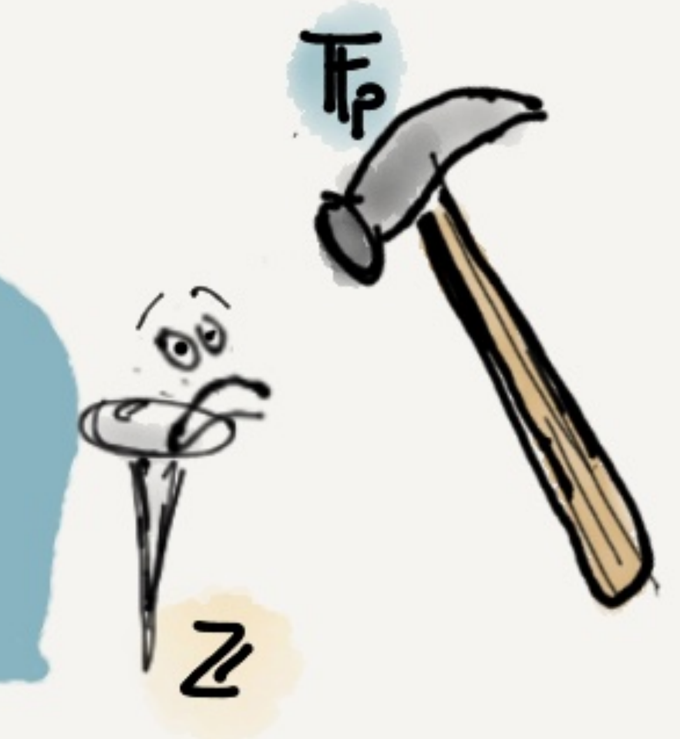
EQUIVALENT TO: "I know $q \in \mathbb{Z}$:

$qN + m = \sigma^e$ " (*)

HOW BIG IS THE CORRESPONDING "CIRCUIT"?

AS A "$\mathbb{Z}$ CIRCUIT"	AS A "$\mathbb{F}$ CIRCUIT"

THE COST OF EMULATING $\mathbb{Z}$ COMPUTATIONS



CASE STUDY: RSA SIGNATURES

$$\sigma^e \equiv m \pmod{N}$$

SIGNATURE MESSAGE PK

e IS TYPICALLY 65537 $(= 2^{16}+1)$

EQUIVALENT TO: "I know $q \in \mathbb{Z}$: $qN + m = \sigma^e$" (*)

HOW BIG IS THE CORRESPONDING "CIRCUIT"?

AS A "$\mathbb{Z}$ CIRCUIT"

16 CONSTRAINTS

AS A "$\mathbb{F}$ CIRCUIT"

THE COST OF EMULATING $\mathbb{Z}$ COMPUTATIONS

CASE STUDY: RSA SIGNATURES

$$\sigma^e \equiv m \pmod{N}$$

SIGNATURE MESSAGE PK

e is TYPICALLY 65537 $(= 2^{16}+1)$

EQUIVALENT TO: "I know $q \in \mathbb{Z}$:

$qN + m = \sigma^e$" (*)

HOW BIG IS THE CORRESPONDING "CIRCUIT"?

AS A "$\mathbb{Z}$ CIRCUIT"

16 CONSTRAINTS

AS A "$\mathbb{F}$ CIRCUIT"

90K CONSTRAINTS

(5000x RATIO!)

WITH STATE OF THE ART COMPILERS [XJSNARK]

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 90K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 90K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS
  - REBUTTAL:
    "WE CAN JUST HAVE A COMPILER DO IT"

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 90K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS
  - REBUTTAL:
    "WE CAN JUST HAVE A COMPILER DO IT"
    - RESPONSE:
      "DO YOU HAVE TIME/RESOURCES TO DESIGN, WRITE, DOCUMENT, MAINTAIN THE COMPILER?"

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 90K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS
  - REBUTTAL: "WE CAN JUST HAVE A COMPILER DO IT"
    - RESPONSE: "DO YOU HAVE TIME/RESOURCES TO DESIGN, WRITE, DOCUMENT, MAINTAIN THE COMPILER?"
  - REBUTTAL: "YES, WE DON'T MIND THE ADDITIONAL COMPLEXITY. ALSO, WE DON'T THINK FINITE FIELDS AND PARAMETERS WILL CHANGE THROUGH TIME"

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 90K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS
  - REBUTTAL: "WE CAN JUST HAVE A COMPILER DO IT"
    - RESPONSE: "DO YOU HAVE TIME/RESOURCES TO DESIGN, WRITE, DOCUMENT, MAINTAIN THE COMPILER?"
      - REBUTTAL: "YES, WE DON'T MIND THE ADDITIONAL COMPLEXITY. ALSO, WE DON'T THINK FINITE FIELDS AND PARAMETERS WILL CHANGE THROUGH TIME"
        - RESPONSE: "THERE IS A BUG IN YOUR CIRCUIT. GOOD LUCK WITH DEBUGGING THE 90K CONSTRAINTS!"

BEYOND EFFICIENCY: SAFER, SIMPLER CIRCUITS (AND HAPPIER DEVELOPERS)

OTHER IMPLICATIONS OF 16 VS 30K CONSTRAINTS:

- NO NEED FOR DEVS TO WRITE/HANDLE LARGE, COMPLEX CIRCUITS
  - REBUTTAL: "WE CAN JUST HAVE A COMPILER DO IT"
    - RESPONSE: "DO YOU HAVE TIME/RESOURCES TO DESIGN, WRITE, DOCUMENT, MAINTAIN THE COMPILER?"
      - REBUTTAL: "YES, WE DON'T MIND THE ADDITIONAL COMPLEXITY. ALSO, WE DON'T THINK FINITE FIELDS AND PARAMETERS WILL CHANGE THROUGH TIME"
        - RESPONSE: "THERE IS A BUG IN YOUR CIRCUIT. GOOD LUCK WITH DEBUGGING THE 30K CONSTRAINTS!"
          - REBUTTAL: "I CAN ALWAYS WRITE A TOOL THAT..."

THIS WORK

ZNARKS: SUCCINCT** (NON-INTERACTIVE) ARGUMENTS THAT WORK NATIVELY* OVER $\mathbb{Z}$

* "NATIVELY" $\approx$ "$\mathbb{Z}$ ARITHMETIC IS A 1ST CLASS CITIZEN AND HAS AS LITTLE OVERHEAD AS POSSIBLE"

** "SUCCINCT": USUAL DEFS + EXTRA (MORE ON THIS LATER)

THIS WORK

ZNARKS: SUCCINCT** (NON-INTERACTIVE) ARGUMENTS THAT WORK NATIVELY* OVER $\mathbb{Z}$

⇒ CONCRETE CONSTRUCTIONS

⇒ GENERAL TECHNIQUES/FRAMEWORK

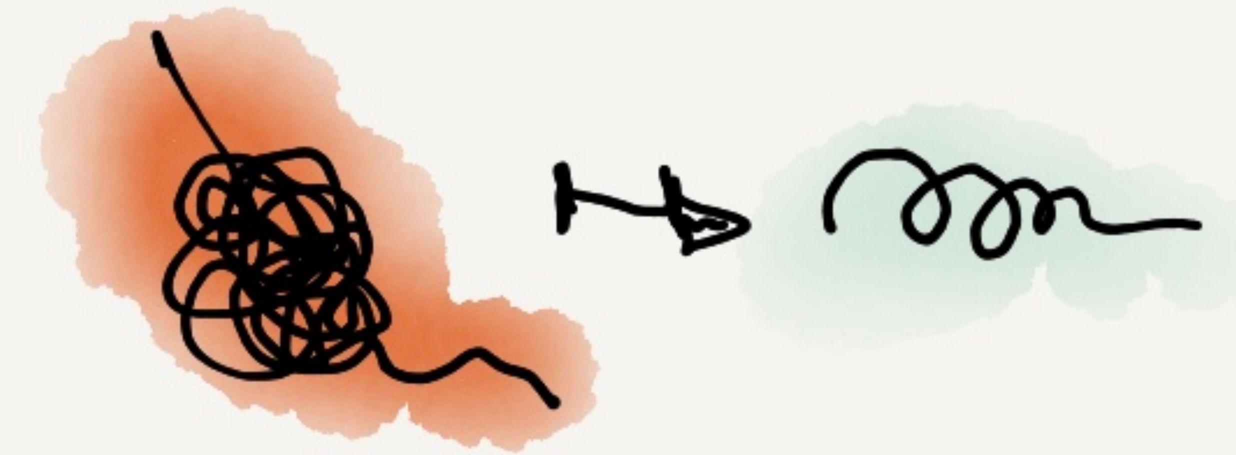
* "NATIVELY" ≈ "$\mathbb{Z}$ ARITHMETIC IS A 1ST CLASS CITIZEN AND HAS AS LITTLE OVERHEAD AS POSSIBLE"

** "SUCCINCT": USUAL DEFS + EXTRA (MORE ON THIS LATER)

NEXT:

MORE ON MOTIVATION

WHY ZNARKS?

- OLD PROBLEMS BECOME EASIER 
  ("WHAT OBVIOUS BOTTLENECKS BECOME LESS SO?")
- SIMPLE TOOLING/GOOD DEV EXP. 
- NEW HORIZONS 
  ("WHAT APPROACHES CAN WE REVISIT (SEMI)ENTIRELY?")

RANGE PROOFS (IN 4 CONSTRAINTS)

FACT: LET $x \in \mathbb{Z}$. $x \geq 0 \iff \exists \alpha, \beta, \gamma, \delta \in \mathbb{Z}$:
$x = \alpha^2 + \beta^2 + \gamma^2 + \delta^2$

(SUM OF 4 SQUARES)

$m \in [low, high] \iff \exists a, b, c$:
$4(m - low)(high - m) + 1 = a^2 + b^2 + c^2$

[Couteau, Peters, Pointcheval]

SNARKS FOR $\mathbb{Q}$

ARITHM. OVER $\mathbb{Z}$ $\Rightarrow$ ARITHM OVER $\mathbb{Z}^2$

- $\underline{\text{MUL}}\left(\underset{\text{in}}{\frac{a}{b}}, \underset{\text{in}}{\frac{c}{d}}, \underset{\text{out}}{\frac{e}{f}}\right) :=$ $\begin{array}{c} e = ac \\ \wedge \\ f = bd \end{array}$

- $\underline{\text{ADD}}$: (SIMILAR)

- To check den $\neq 0$: ADAPT TECHNIQUES FROM PREV. SLIDE

ARBITRARY PRECISION ARITHMETIC

SNARKS FOR $\mathbb{Q}$ $\Longrightarrow$ SNARKS FOR "DECIMAL NUMBERS"

POTENTIAL APPLICATIONS:

SCIENTIFIC COMPUTATION,
MATH STATEMENTS,
(AND MORE)

NEW APPROACHES/AVENUES FOR SNARKS FOR ML

TWO AVENUES

1) DIRECT:

[LUO & SUN '24] "ADDITION IS ALL YOU NEED FOR ENERGY EFFICIENT LANGUAGE MODELS"

2) INDIRECT:

- ARB. PRECISION?
- NEW APPROACHES TO FLOATING-POINT ARITHM?

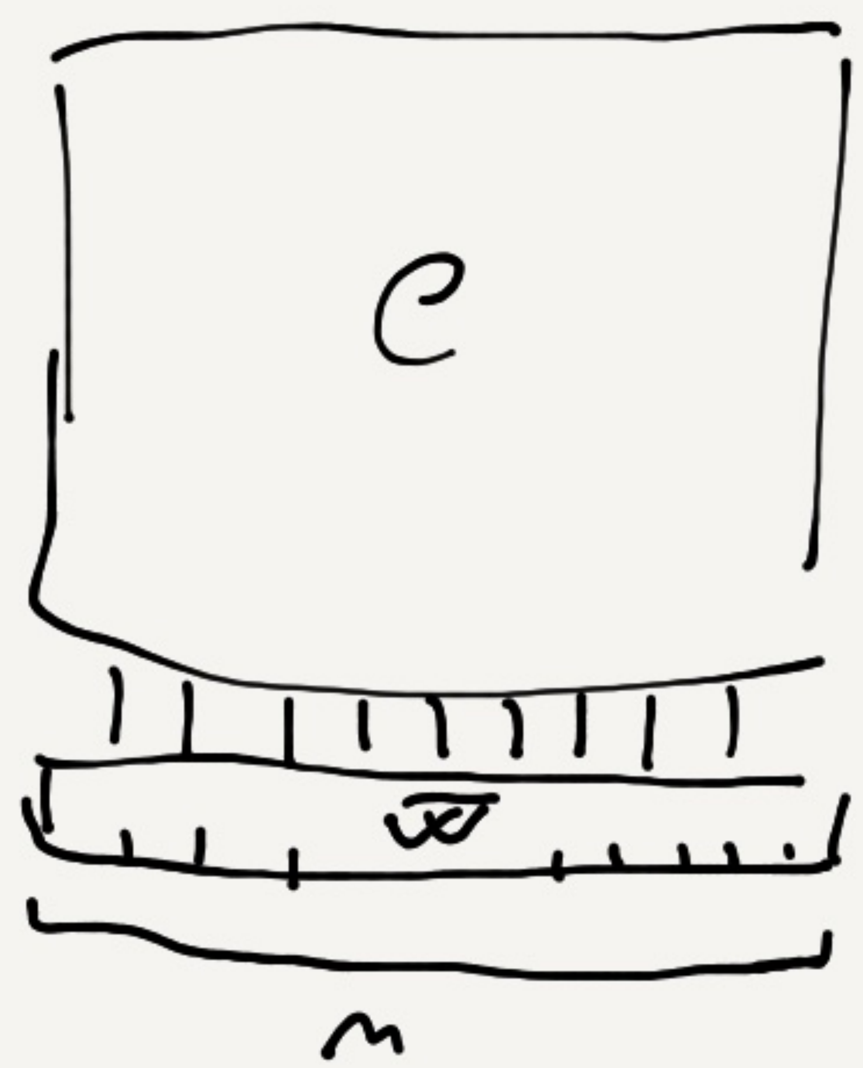
BUT ENOUGH ABOUT MOTIVATION.

WHAT ABOUT THIS WORK?

(ALMOST THERE, BUT FIRST:

- A NOTE ON SUCCINCTNESS
- PRIOR WORK)

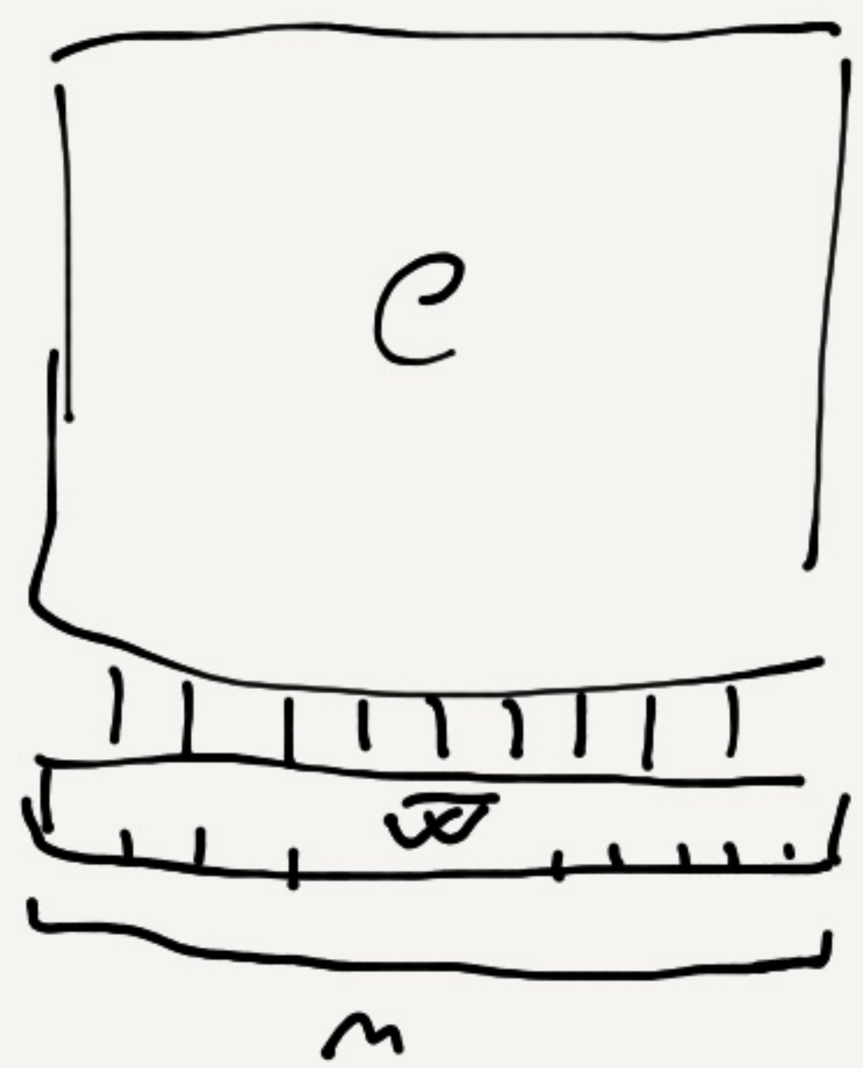
"SUCCINCTNESS" FOR Z COMPUTATIONS



SUCCINCTNESS (TRADITIONALLY) := $|\pi|_{C}$, $\mathrm{TIME}(V)_{C}$ $o(|w|)$

SMALL OH

"SUCCINCTNESS" FOR $\mathbb{Z}$ COMPUTATIONS



SUCCINCTNESS (TRADITIONALLY) :=

$$\left.\begin{array}{l} |\pi| \\ \mathrm{TIME}(V) \end{array}\right\} \in o(|w|)$$

SMALL OH

BUT, OVER $\mathbb{Z}$, WHAT ABOUT

$\bar{w} := (2^{60} - 42, 50!, \text{SOME\_MERSENNE\_PRIME})$??

$|\bar{w}| = 3$

BUT $\|\bar{w}\|_\infty$ IS HUGE

WE WANT SUCCINCTNESS IN BOTH $|w|$ AND $\|w\|_\infty$ (FULL SUCCINCTNESS)

PRIOR WORK

SNARKS OVER RINGS:

RINOCCHIO
(GANESH et AL., JoC'23)

- DESIGNATED-VERIFIER
- (RELATIVELY) UNSTUDIED ASSUMPTIONS (KOE-STYLE) OVER RINGS

"GKR FOR INFINITE, NON-COMM. RINGS"
(SORIA-VAZQUEZ, TCC'22)

- DETERMINISTIC COMPUTATIONS ONLY
- WAY BEYOND $\mathbb{Z}$

PRIOR WORK

SNARKS OVER RINGS:

RINOCCHIO
(GANESH et al., JoC'23)

"GKR FOR INFINITE, NON-COMM. RINGS"
(SORIA-VAZQUEZ, TCC'22)

- DESIGNATED-VERIFIER
- (RELATIVELY) UNSTUDIED ASSUMPTIONS (KOE-STYLE) OVER RINGS
- DETERMINISTIC COMPUTATIONS ONLY
- WAY BEYOND $\mathbb{Z}$

SUCCINCT DIOPHANTINE-SAT. ARGUMENTS

(TOWA & VERGNAUD, AC'20)

- "BULLETPROOFS OVER $\mathbb{Z}$ VIA INTEGER COMMITMENTS"
- NOT FULLY SUCCINCT ($|\pi|$ IS $O(\log|w| + \|w\|_\infty)$)
- VERIFIER NOT SUCCINCT (LINEAR)
- HAS A (RESTRICTED) FORM OF ZK (WE HAVE NO ZK)

LEAKS $\|w\|_\infty$

PRIOR WORK

SNARKs OVER RINGS:

RINOCCHIO (GANESH et al., JoC '23) — PINOCCHIO

- DESIGNATED-VERIFIER
- (RELATIVELY) UNSTUDIED ASSUMPTIONS (KOE-STYLE) OVER RINGS

"GKR FOR INFINITE, NON-COMM. RINGS" (SORIA-VAZQUEZ, TCC '22) — GKR

- DETERMINISTIC COMPUTATIONS ONLY
- WAY BEYOND $\mathbb{Z}$

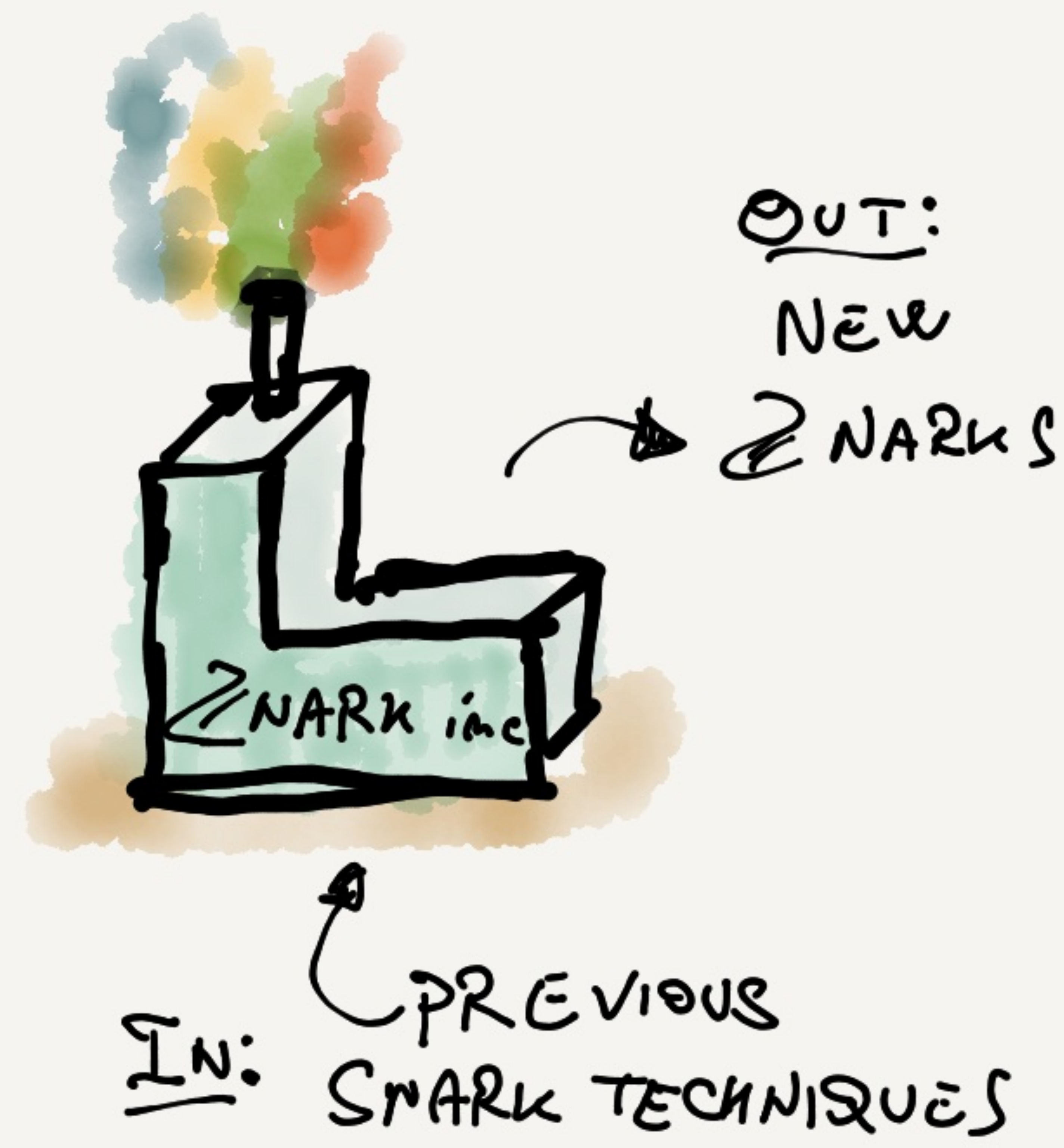
SUCCINCT DIOPHANTINE-SAT. ARGUMENTS (TOWA & VERGNAUD, AC '20) — BP

- "BULLETPROOFS OVER $\mathbb{Z}$ VIA INTEGER COMMITMENTS"
- NOT FULLY SUCCINCT ($|\pi|$ IS $O(\log|w| + \|w\|_\infty)$)
- VERIFIER NOT SUCCINCT (LINEAR)
- HAS A (RESTRICTED) FORM OF ZK (WE HAVE NO ZK)
  - LEAKS $\|w\|_\infty$

ALSO, ALL PRIOR WORKS ARE SOMEWHAT SCHEME-SPECIFIC (OUR GOAL: GENERAL TREATMENT)

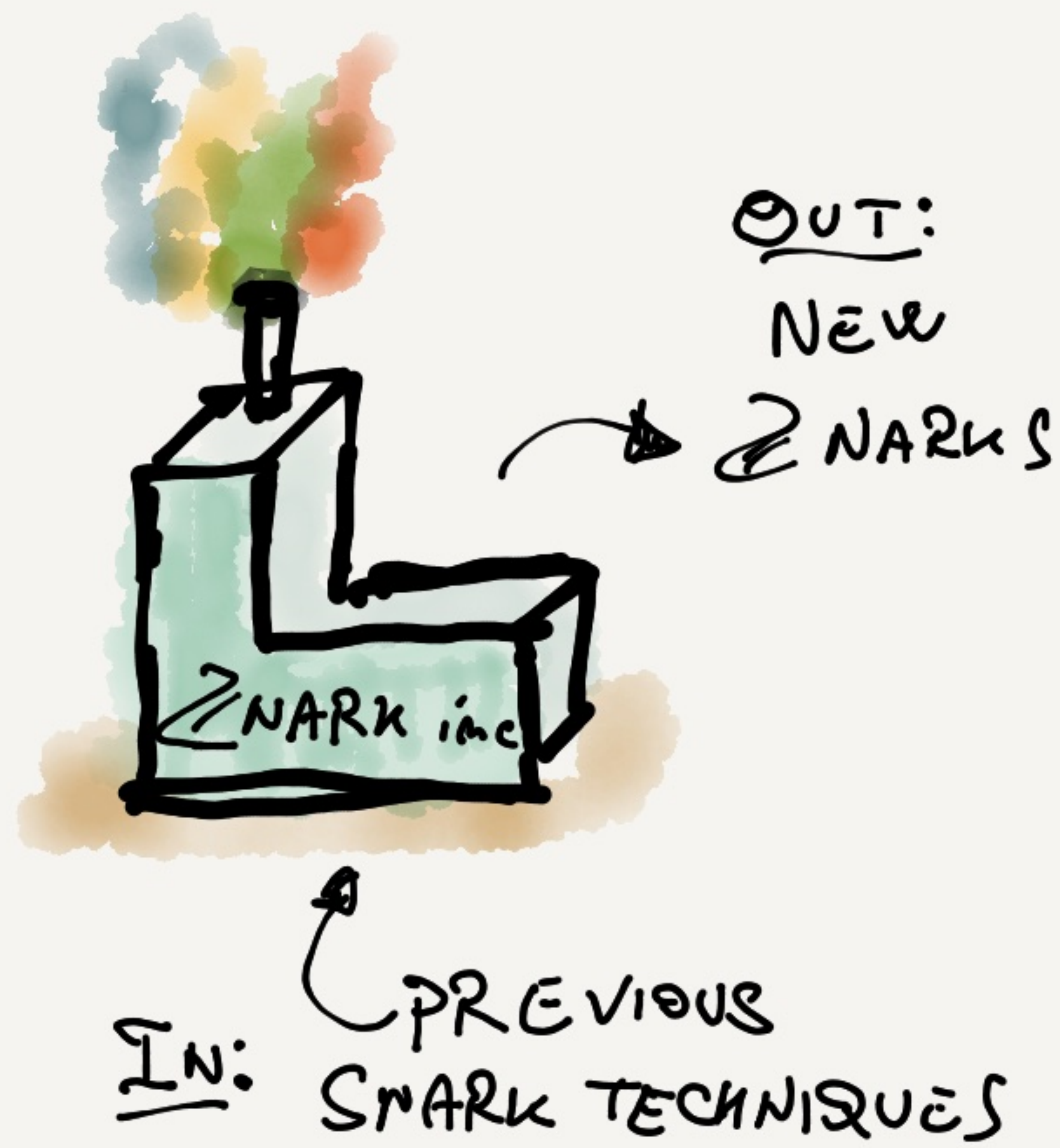
THIS WORK

TECHNIQUES
&
FRAMEWORK
FOR
(FULLY SUCCINCT) ZNARKS



THIS WORK

TECHNIQUES & FRAMEWORK FOR (FULLY SUCCINCT) ZNARKS



BUILDING BLOCKS/INSTANTIATIONS

THIS WORK

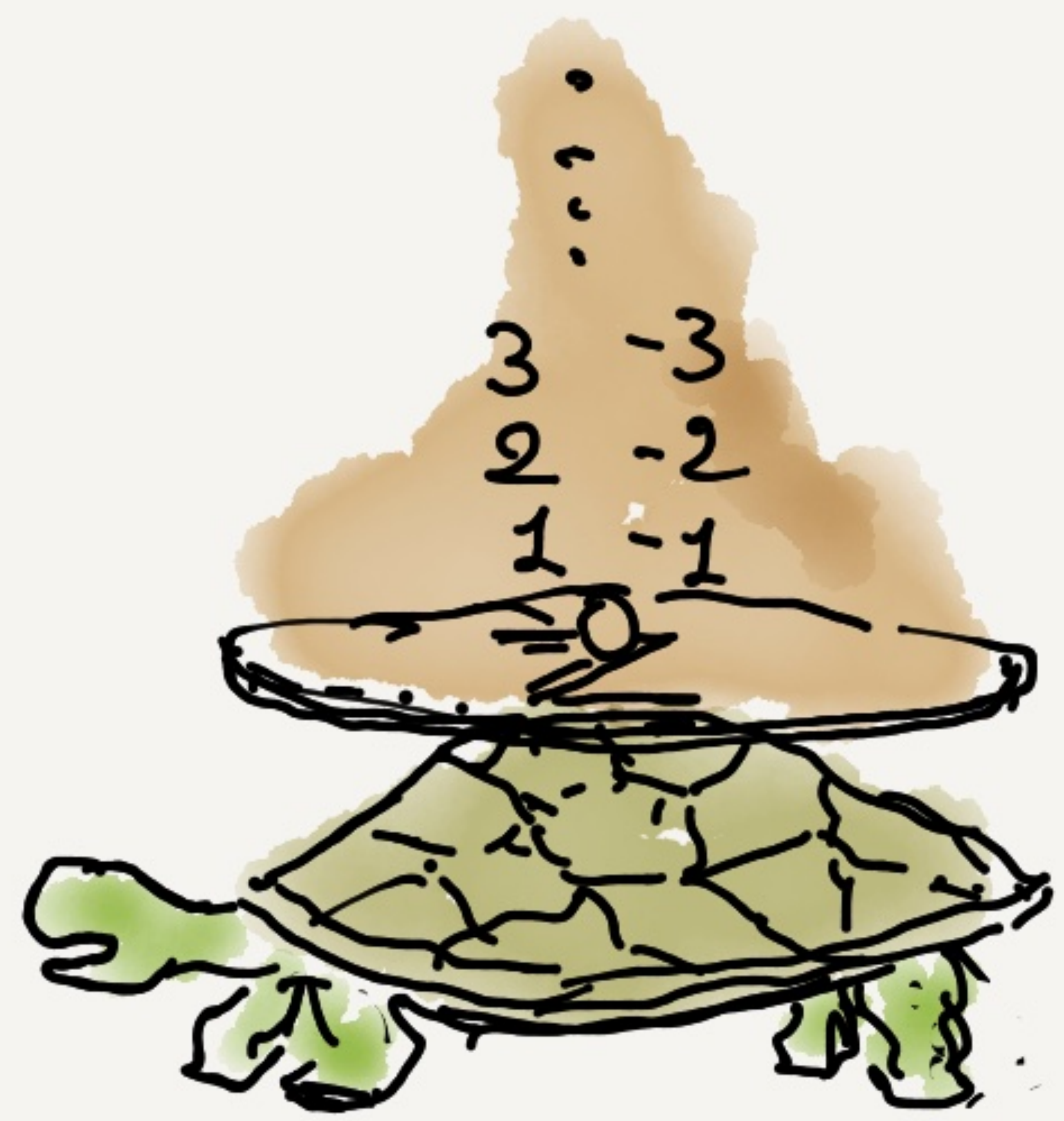
TECHNIQUES & FRAMEWORK FOR (FULLY SUCCINCT) ZNARKS

OUT: NEW ZNARKS

IN: PREVIOUS SNARK TECHNIQUES

BUILDING BLOCKS/INSTANTIATIONS 

CONCRETE CONSTRUCTION : ZARATAN (SPARTAN FOR Z)



THIS WORK

TECHNIQUES & FRAMEWORK FOR (FULLY SUCCINCT) $\mathbb{Z}$NARKS

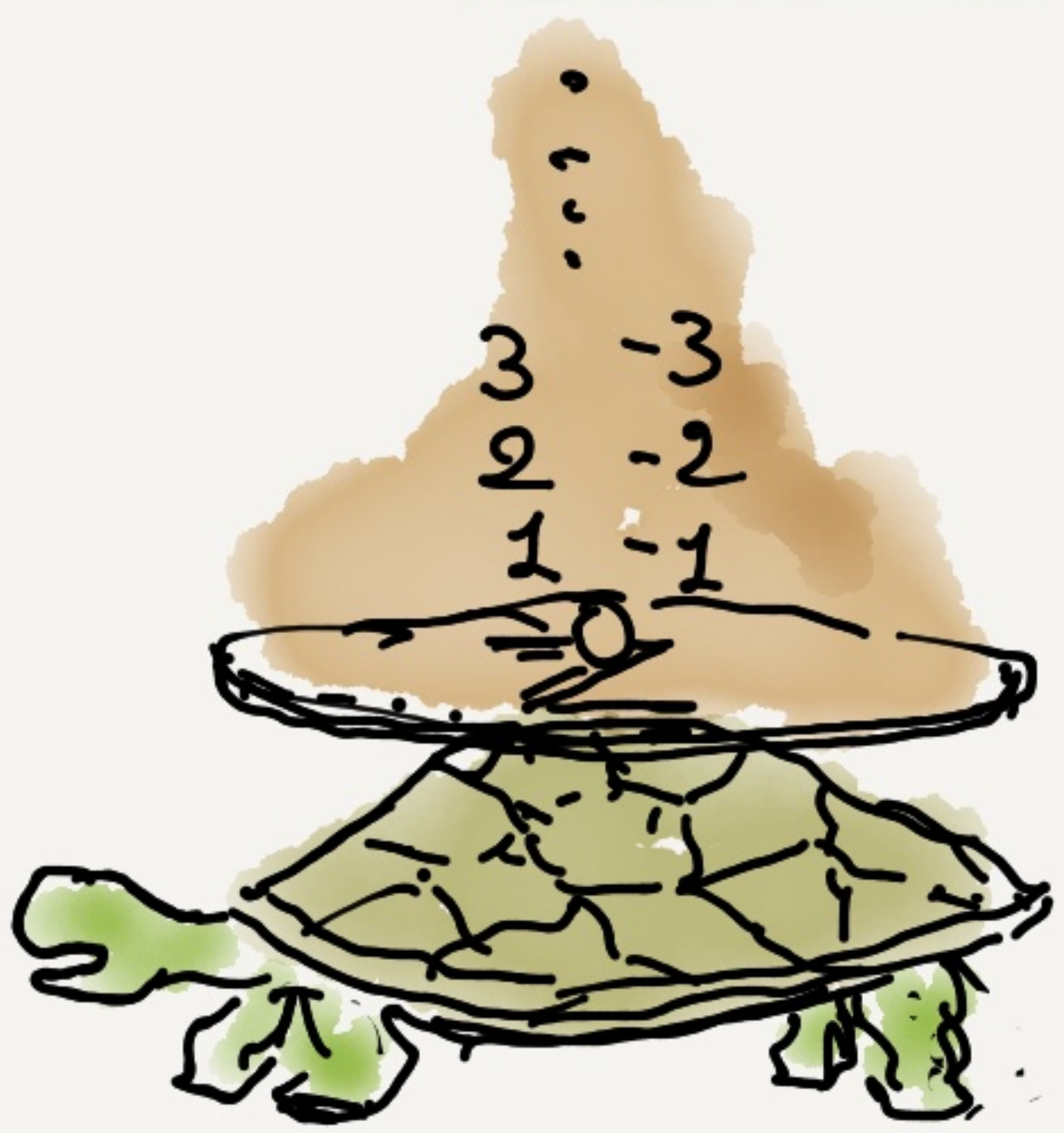
OUT: NEW $\mathbb{Z}$NARKS

IN: PREVIOUS SNARK TECHNIQUES

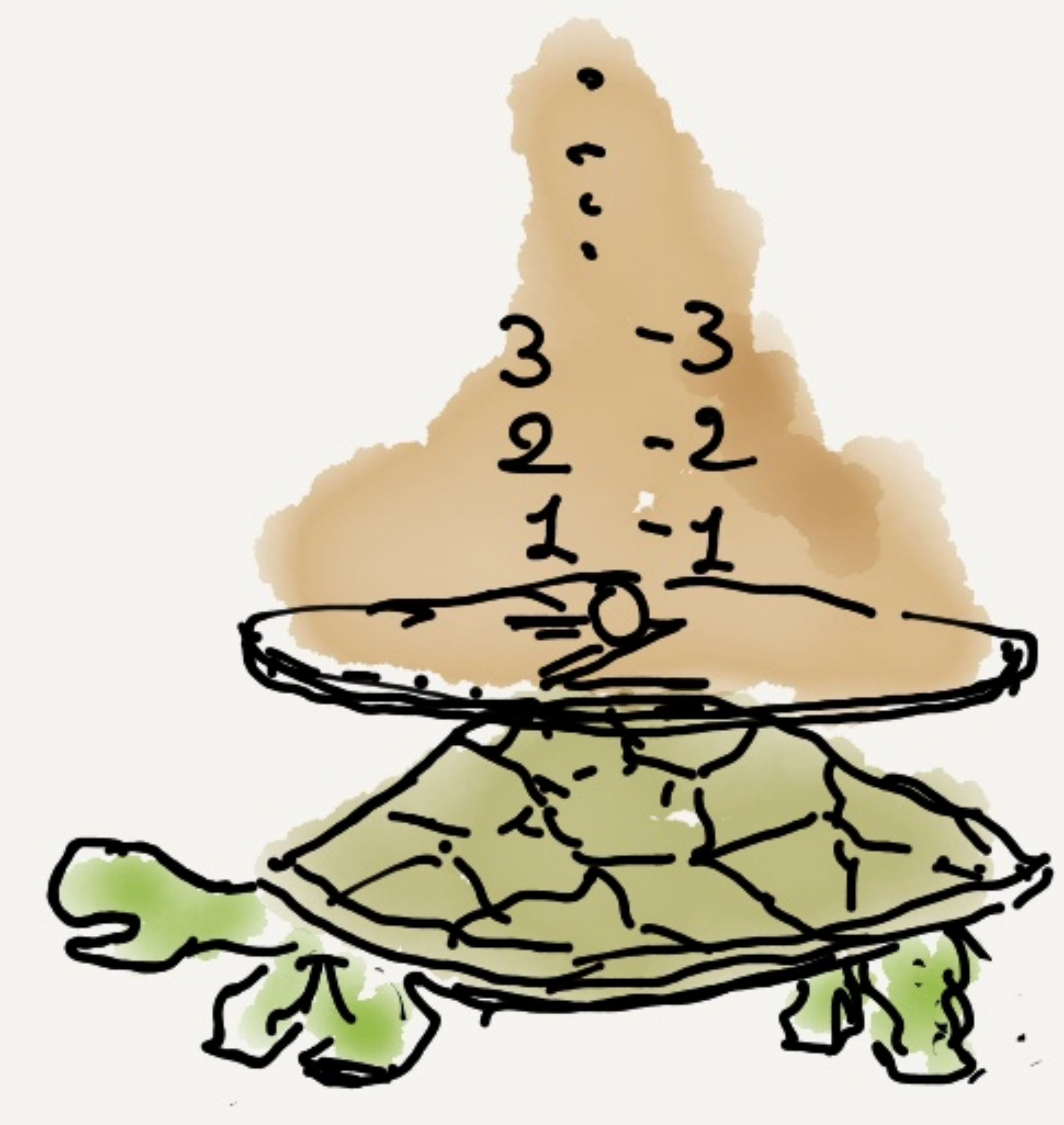
BUILDING BLOCKS/INSTANTIATIONS

CONCRETE CONSTRUCTION : $\mathbb{Z}$ARATAN (SPARTAN FOR $\mathbb{Z}$)

FIRST FULLY SUCCINCT SNARK FOR $\mathbb{Z}$



EFFICIENCY OF ZARATAN



$\vec{w} \in \mathbb{Z}^n$

n: # OF R1CS $\mathbb{Z}$ CONSTRAINTS

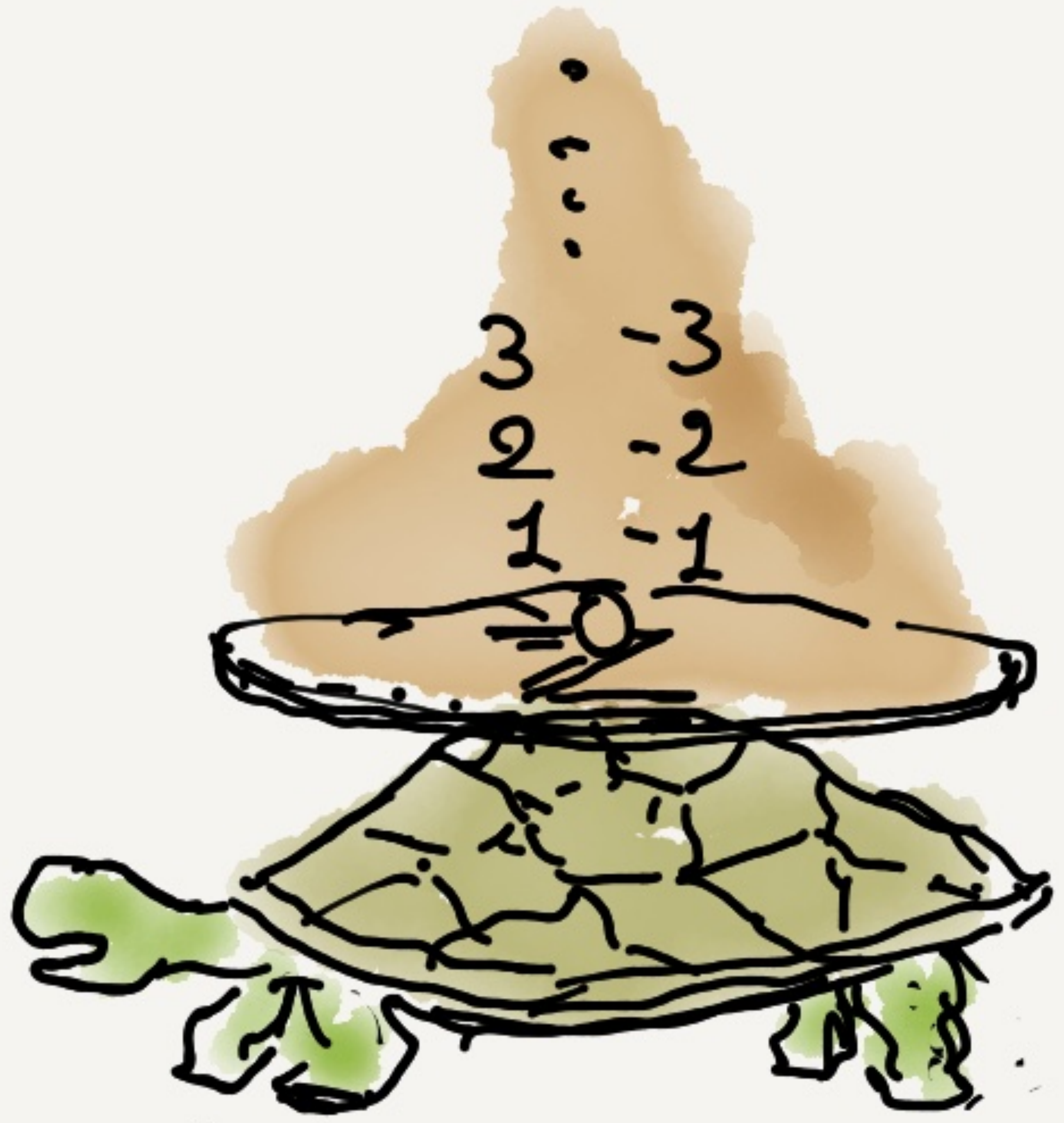
m: MAX-SIZE IN BITS OF EACH $|w_i|$

<u>NB</u>: $|\text{COMPUTATION}| = n \cdot m$

EFFICIENCY OF ZARATAN

TIME(V) IS SIMILAR

	$\|\pi\|$	ANY COMPUTATION?	TIME(P)
FULLY SUCCINCT	$\log(m)$	SOME: $\frac{m^2}{n} < 1$	LINEAR
ZARATAN	$+ \lg^2(n)$	ANY	"LINEAR--"



1 STEP IN 1 SUPROTOCOL IN 1 OF THE TWO COMPONENTS OF ZARATAN REQUIRES $\approx m^2$ BIT OPS

$\overline{w} \in \mathbb{Z}^n$

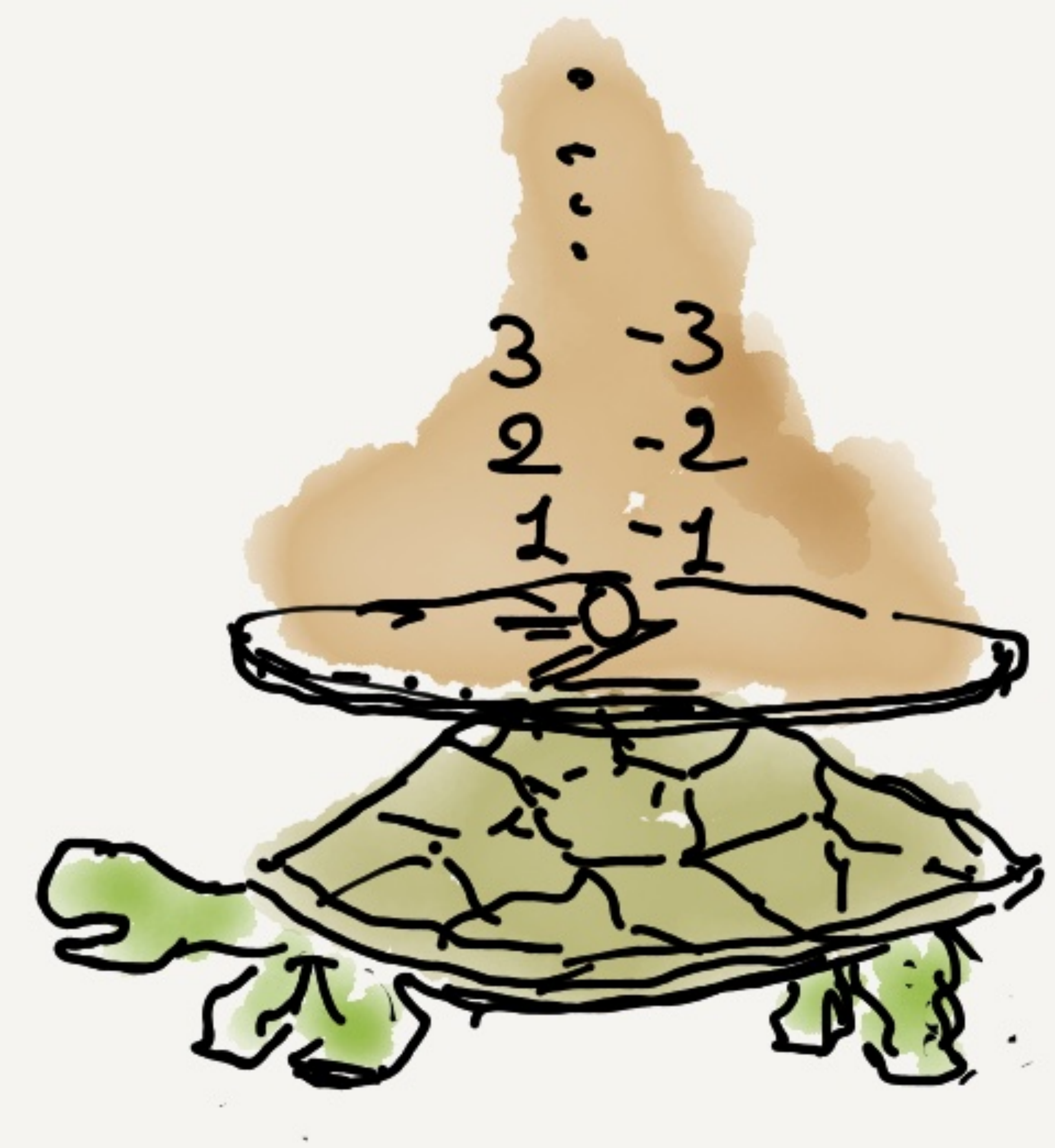
n : # OF R1CS $\mathbb{Z}$ CONSTRAINTS

m : MAX-SIZE IN BITS OF EACH $|w_i|$

NB: $|\text{COMPUTATION}| = n \cdot m$

EFFICIENCY OF ZARATAN

	$\lvert\pi\rvert$	ANY COMPUTATION?	TIME(P)
FULLY SUCCINCT ZARATAN	$\log(m)$ + $\lg^2(m)$	SOME: $\frac{m^2}{n} < 1$	LINEAR
		ANY	"LINEAR--"
SUCCINCT ZARATAN	(ABOVE) + $\approx m$	ANY	LINEAR



↳ 1 STEP IN
1 SUPROTOCOL IN
1 OF THE TWO COMPONENTS
OF ZARATAN
REQUIRES $\approx m^2$ BIT OPS

$\overline{w} \in \mathbb{Z}^n$

n : # OF R1CS
$\mathbb{Z}$ CONSTRAINTS

m : MAX-SIZE IN
BITS OF EACH $\lvert w_i \rvert$

NB: $\lvert \text{COMPUTATION} \rvert = n \cdot m$

OVERVIEW OF TECHNIQUES

Our Core Technique:

Fingerprinting

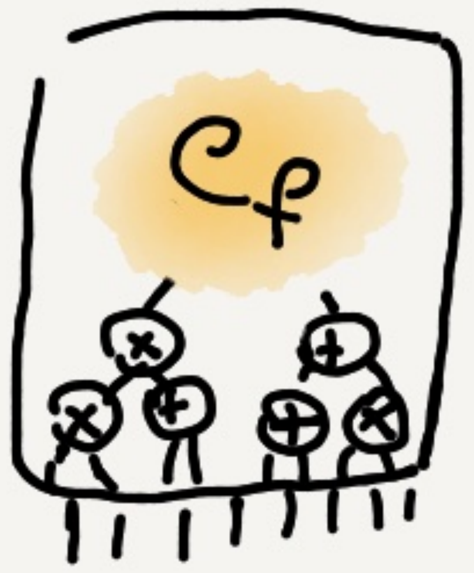
FINGERPRINTING THROUGH AN EXAMPLE:

ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



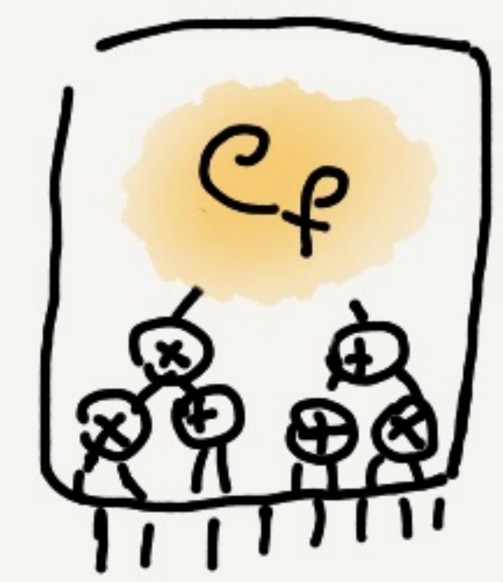
$|C_f| = \mathrm{poly}(m)$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = \mathrm{poly}(m)$

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

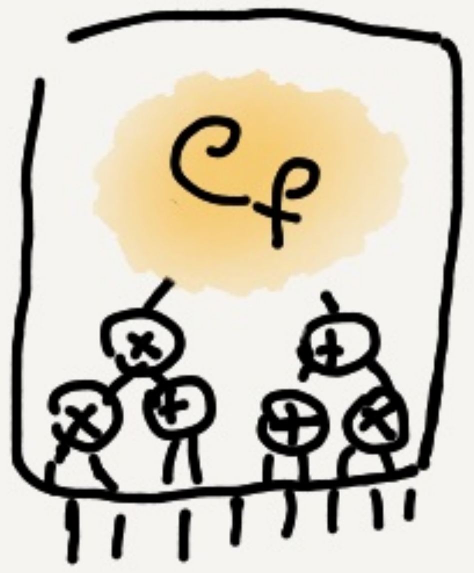
$$\Pr_{r_1 \dots r_m \leftarrow S}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = \mathrm{poly}(m)$

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

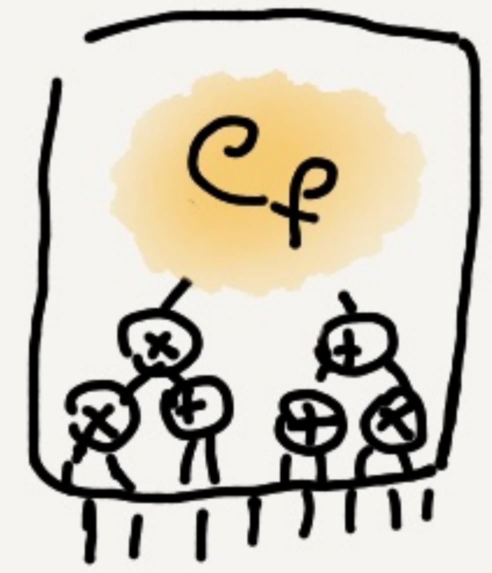
$$\Pr_{r_1 \dots r_m \xleftarrow{\$} S}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE:
ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = poly(m)$

$\Downarrow$

OBS: $deg(f) = O(2^{m^c})$

(WHY?

- EACH $\otimes \Rightarrow$ DEGREE DOUBLES
- POLY DEPTH $\Rightarrow 2^{poly}$ DEGREE)

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

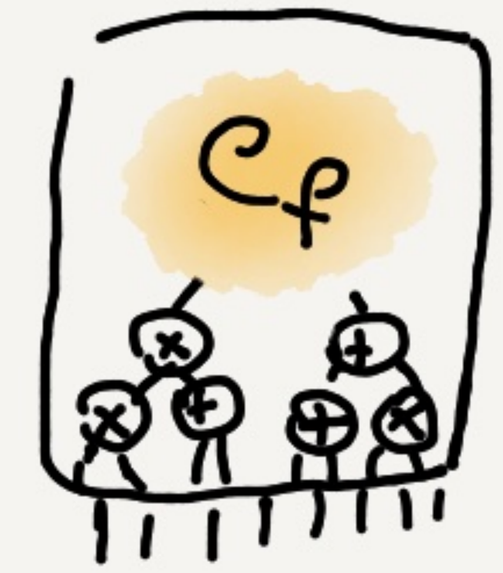
$$\Pr_{r_1 \dots r_m \xleftarrow{\$} S}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = poly(m)$

$\Downarrow$

OBS: $\deg(f) = O(2^{m^c})$

(WHY?
- EACH $\otimes \Rightarrow$ DEGREE DOUBLES
- POLY DEPTH $\Rightarrow 2^{poly}$ DEGREE)

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$

WHAT M? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS

$\hookrightarrow \log(M) = poly$

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

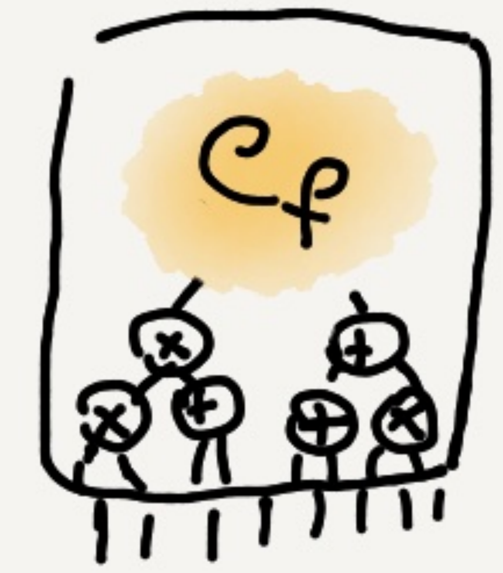
$$\Pr_{\substack{r_1 \dots r_m \\ \leftarrow S}} \left[f(r_1, \dots, r_m) \neq 0 \right] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = poly(m)$

$\Downarrow$

OBS: $\deg(f) = O(2^{m^c})$

(WHY?
- EACH $\otimes \Rightarrow$ DEGREE DOUBLES
- POLY DEPTH $\Rightarrow 2^{poly}$ DEGREE)

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$

WHAT M? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS

$\hookrightarrow \log(M) = poly$

BUT... NOT POLY TIME!! WHY?

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

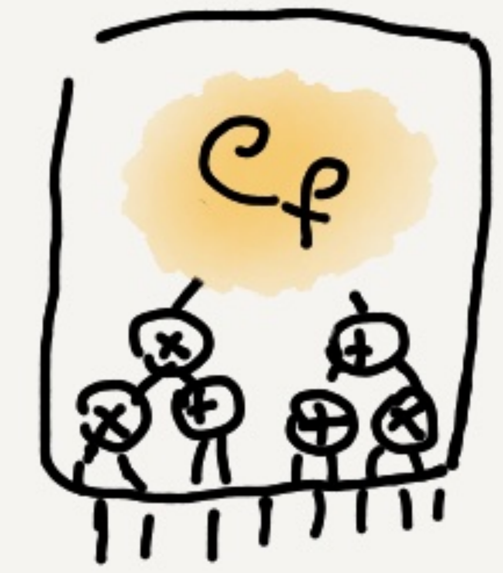
$$\Pr_{r_1 \dots r_m \xleftarrow{\$} S}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



$|C_f| = \text{poly}(m)$

$\Downarrow$

OBS: $\deg(f) = O(2^{m^c})$

(WHY?
- EACH $\otimes \Rightarrow$ DEGREE DOUBLES
- POLY DEPTH $\Rightarrow 2^{\text{poly}}$ DEGREE)

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$ ✓ ✓

WHAT M? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS

↳ $\log(M) = \text{poly}$ ✓

BUT... NOT POLYTIME!! WHY?

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

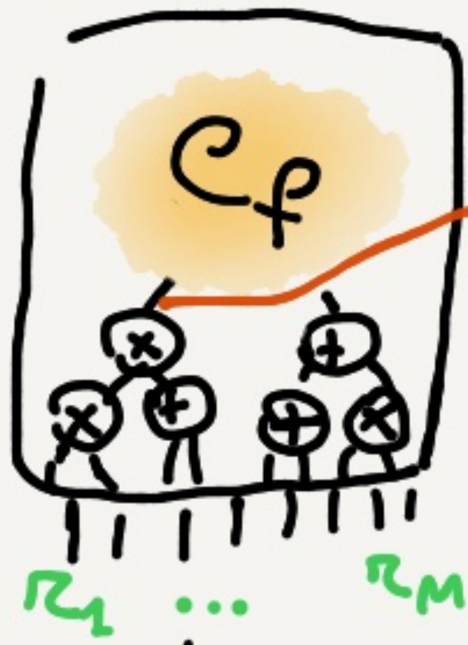
$$\Pr_{r_1 \dots r_m \xleftarrow{\$} S}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_m]$



INTERMEDIATE WIRES NOT POLY

$|C_f| = \text{poly}(m)$

$\Downarrow$

OBS: $\deg(f) = O(2^{m^c})$

(WHY?

- EACH $\otimes \Rightarrow$ DEGREE DOUBLES
- POLY DEPTH $\Rightarrow 2^{\text{poly}}$ DEGREE)

IMPLIES BIT SIZE DOUBLES

APPROACH: SAMPLE $r_1, \dots, r_m \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)

return $f(r_1, \dots, r_m) \stackrel{?}{=} 0$

WHAT M? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS

$\hookrightarrow \log(M) = \text{poly}$ ✓

BUT... NOT POLYTIME!! WHY?

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1 \dots x_m]$, $f \neq 0$:

$$\Pr_{\substack{r_1 \dots r_m \\ \leftarrow S}}\left[f(r_1, \dots, r_m) \neq 0\right] \geq 1 - \frac{\deg(f)}{|S|}$$

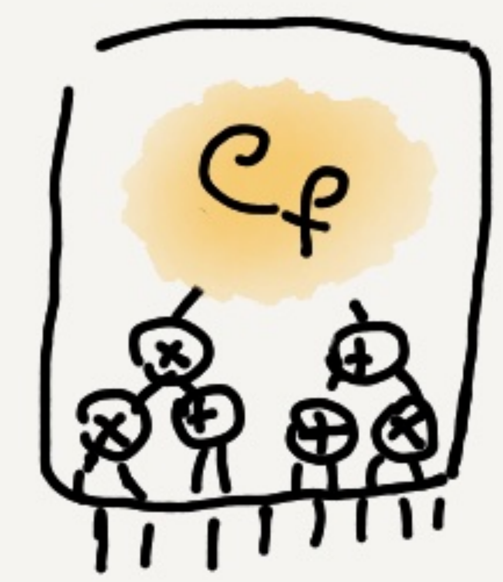
FINGERPRINTING THROUGH AN EXAMPLE:

ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \ldots, x_n]$



$|C_f| = \text{poly}(n)$

$\Downarrow$

$\deg(f) = O\left(2^{n^c}\right)$

APPROACH:
SAMPLE $r_1, \ldots, r_n \leftarrow \{1, \ldots, M\}$ // $M \approx 100 \cdot \deg(f)$
return $f(r_1, \ldots, r_n) \stackrel{?}{=} 0$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

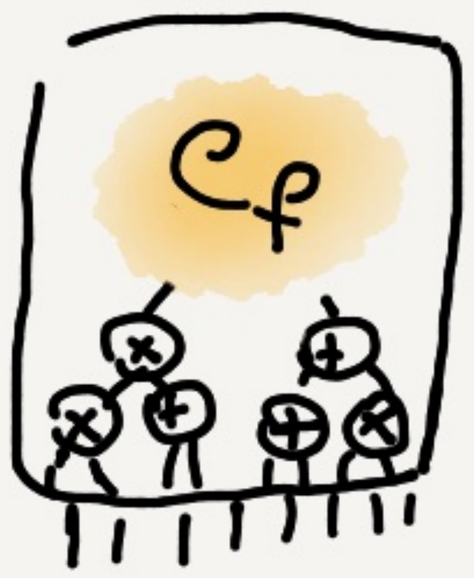
$f \in \mathbb{Z}[x_1, \dots, x_n]$

APPROACH:

SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ // $M \approx 100 \cdot \deg(f)$

return $\underline{f(r_1, \dots, r_n) \overset{?}{=} 0 \pmod{q}}$

(FOR RANDOM PRIME OF ℓ BITS)



$|C_f| = \mathrm{poly}(n)$

$\Downarrow$

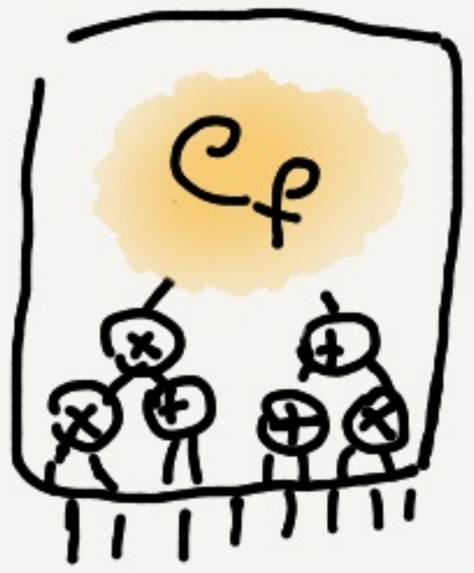
$\deg(f) = O(2^{n^c})$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_n]$



$|C_f| = poly(n)$

$\Downarrow$

$\deg(f) = O(2^{n^c})$

APPROACH:
SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ // $M \approx 100 \cdot \deg(f)$

return $f(r_1, \dots, r_n) \stackrel{?}{=} 0 \pmod q$

(FOR RANDOM PRIME OF ℓ BITS)

WHY DOES THIS WORK?

- IF $f(r_1, \dots, r_n) = 0$ THEN $f(r_1, \dots, r_n) \equiv 0 \pmod q$ ✓

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

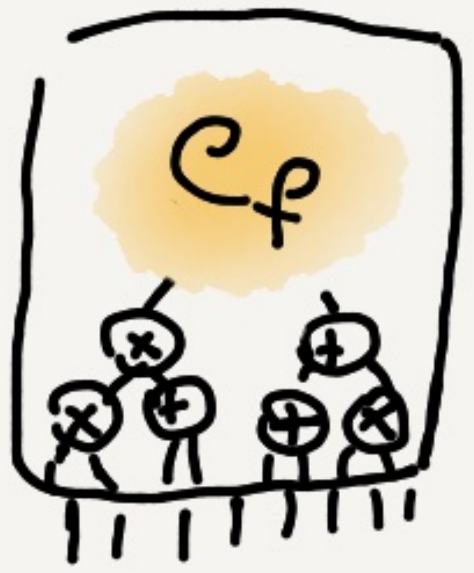
OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \dots, x_n]$

APPROACH:
SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ // $M \approx 100 \cdot \deg(f)$

return $f(r_1, \dots, r_n) \stackrel{?}{=} 0 \pmod q$

(FOR RANDOM PRIME OF ℓ BITS)



$|C_f| = \mathrm{poly}(n)$

$\Downarrow$

$\deg(f) = O(2^{n^c})$

WHY DOES THIS WORK?

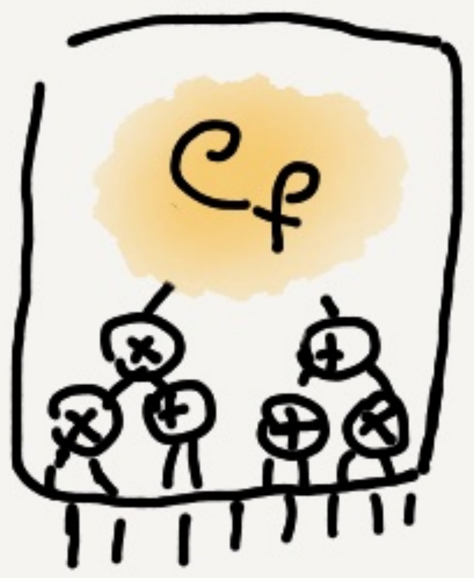
- IF $f(r_1, \dots, r_n) = 0$ THEN $f(r_1, \dots, r_n) \equiv 0 \pmod q$ ✓
- IF $\underbrace{f(r_1, \dots, r_n)}_{y} \neq 0$ THEN $f(r_1, \dots, r_n) \not\equiv 0 \pmod q$ UNLESS $q | y$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

$f \in \mathbb{Z}[x_1, \ldots, x_m]$



$|C_f| = \mathrm{poly}(m)$

$\Downarrow$

$\deg(f) = O\left(2^{m^c}\right)$

APPROACH:
- SAMPLE $r_1, \ldots, r_m \leftarrow \{1, \ldots, M\}$ // $M \approx 100 \cdot \deg(f)$
- return $f(r_1, \ldots, r_m) \stackrel{?}{=} 0 \pmod q$

(FOR RANDOM PRIME OF ℓ BITS)

WHY DOES THIS WORK?

- IF $f(r_1, \ldots, r_m) = 0$ THEN $f(r_1, \ldots, r_m) \equiv 0 \pmod q$ ✓
- IF $\underbrace{f(r_1, \ldots, r_m)}_{y} \neq 0$ THEN $f(r_1, \ldots, r_m) \not\equiv 0 \pmod q$

UNLESS $q | y$

equivalent to:

$q \in \{p_1 \ldots p_t\}$ w/ $y = p_1^{\ell_1} \cdots p_t^{\ell_t}$

WE CAN BOUND THE $\Pr[\ ]$ OF THIS EVENT BY PRIME NUMBERS THM AND $|q| \approx \mathrm{poly}$

FINGERPRINTING FOR OUR SETTING*

(* NON-DETERMINISTIC COMPUTATIONS OVER $\mathbb{Z}$)

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$

VERIFIER (vk_C)

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$

VERIFIER (vk_C)

INTUITION(S) WE WILL TRY* AND LEVERAGE:

* AND IN GENERAL THIS WILL FAIL
(PLUS WILL STILL REQUIRE MORE WORK)

AN ATTEMPT TO A TEMPLATE

$\mathcal{C}(\bar{w})=0$

LET $\mathcal{C}$ BE <u>SOME</u> "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$ — **VERIFIER** $(vk_{\mathcal{C}})$

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![\mathcal{C}]\!]_q :=$ "$\mathcal{C}$ mod q"

$\pi \leftarrow$ SOME SNARK.PROVE$\big(q,$ "$\exists\, \bar{w}$:
- $[\![\mathcal{C}]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm" $\big)$

$\xrightarrow{\quad \pi \quad}$

$Vfy\big(q, [\![vk]\!]_q, cm, \pi\big)$

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER$(\bar{w} \in \mathbb{Z}^m)$ VERIFIER(vk_C)

$cm \leftarrow \mathrm{Com}(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q :=$ "$C \bmod q$"

$\pi \leftarrow$ SOMESNARK.PROVE$($q$,$ "$\exists\, \bar{w}$:
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm"$)$

$\xrightarrow{\quad \pi \quad}$

$\mathrm{Vfy}(q, [\![vk]\!]_q, cm, \pi)$

BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOMESNARK"? WHAT PROPERTIES NECESSARY?
2) "$[\![C]\!]_q$" MEANS...? (AND IS IT SOUND?)
3) GENERATING π, HOW TO SWITCH FROM $\mathbb{Z}$ (cm) TO A COMPUTATION mod q?

: BEFORE SAMPLING q
: AFTER SAMPLING q

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$

VERIFIER (vk_C)

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{cm}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{q}$

$[\![C]\!]_q$:= "C mod q"

$\pi \leftarrow$ SOME SNARK.PROVE$(q,$ "$\exists\, \bar{w}$:

- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ is COMMITTED IN cm" $)$

$\xrightarrow{\pi}$

$Vfy(q, [\![vk]\!]_q, cm, \pi)$

BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOMESNARK"? WHAT PROPERTIES NECESSARY?

CHALLENGES:

- q IS RANDOM: HOW TO ENSURE {DLOG-, FFT-, UNIVARIATE-SUMCHECK}-FRIENDLINESS?

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER$(\bar{w} \in \mathbb{Z}^m)$ VERIFIER(vk_C)

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q$:= "C mod q"

$\pi \leftarrow$ SOME SNARK.PROVE$\Big(q$, "$\exists\, \bar{w}$:
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm" $\Big)$

$\xrightarrow{\quad \pi \quad}$ $Vfy\big(q, [\![vk]\!]_q, cm, \pi\big)$

BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOMESNARK"? WHAT PROPERTIES NECESSARY?

CHALLENGES:
- q IS RANDOM: HOW TO ENSURE {DLOG-, FFT-, UNIVARIATE-SUMCHECK}-FRIENDLINESS?
- FIELD-AGNOSTIC WON'T HELP (q IS UNKNOWN FOR "TOO LONG")

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$ | VERIFIER (vk_C)

$cm \leftarrow \mathrm{Com}(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q :=$ "$C \bmod q$"

$\pi \leftarrow$ SOME SNARK.PROVE (q, "$\exists\, \bar{w}$:
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm")

$\xrightarrow{\quad \pi \quad}$

$\mathrm{Vfy}(q, [\![vk]\!]_q, cm, \pi)$

BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOMESNARK"? WHAT PROPERTIES NECESSARY?

CHALLENGES:
- q IS RANDOM: HOW TO ENSURE {DLOG-, FFT-, UNIVARIATE-SUMCHECK}-FRIENDLINESS?
- FIELD-AGNOSTIC WON'T HELP (q IS UNKNOWN FOR "TOO LONG")

SCHEMES BASED ON MLE GOOD CANDIDATES (SPARTAN, HYPERPLONK, ...)

↳ MULTI-LINEAR EXTENSIONS

AN ATTEMPT TO A TEMPLATE

$C(\overline{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\overline{w} \in \mathbb{Z}^m)$

VERIFIER (vk_C)

$cm \leftarrow Com(\overline{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q :=$ "$C \bmod q$"

$\pi \leftarrow$ SOME SNARK.PROVE$\big(q,$ "$\exists\, \overline{w}$:

- $[\![C]\!]_q(\overline{w} \bmod q) = 0$
- $\overline{w}$ IS COMMITTED IN cm" $\big)$

$\xrightarrow{\quad \pi \quad}$

$Vfy\big(q, [\![vk]\!]_q, cm, \pi\big)$

BUT, WE ARE FAR FROM BEING DONE:

1) "$[\![C]\!]_q$" MEANS...? (AND IS IT SOUND?)

$\mathbb{Z}$-R1CS: $\{(\langle \overline{a}_i, \overline{w}\rangle)\circ(\langle \overline{b}_i, \overline{w}\rangle) - \langle \overline{c}_i, \overline{w}\rangle = 0\}_i$

SPARSE VECTOR OF INTEGERS

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$

VERIFIER (vk_C)

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{cm}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{q}$

$[\![C]\!]_q :=$ "C mod q"

$\pi \leftarrow$ SOME SNARK.PROVE $\left(q, \text{"}\exists\, \bar{w}: \right.$
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm"$\left.\right)$

$\xrightarrow{\pi}$

$Vfy\left(q, [\![vk]\!]_q, cm, \pi\right)$

BUT, WE ARE FAR FROM BEING DONE:

1) "$[\![C]\!]_q$" MEANS...? (AND IS IT SOUND?)

$\mathbb{Z}$-R1CS: $\left\{ (\langle \bar{a}_i, \bar{w}\rangle) \circ (\langle \bar{b}_i, \bar{w}\rangle) - \langle \bar{c}_i, \bar{w}\rangle = 0 \right\}_i$

SPARSE VECTOR OF INTEGERS

$\left\{ (\langle \bar{a}_i, \bar{w}\rangle) \circ (\langle \bar{b}_i, \bar{w}\rangle) - \langle \bar{c}_i, \bar{w}\rangle \equiv 0 \pmod q \right\}_i$

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$ | **VERIFIER** (vk_C)

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q :=$ "C mod q"

$\pi \leftarrow$ SOME SNARK.PROVE$\Big(q,$ "$\exists\, \bar{w}$:
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm" $\Big)$

$\xrightarrow{\quad \pi \quad}$

$Vfy\big(q, [\![vk]\!]_q, cm, \pi\big)$

BUT, WE ARE FAR FROM BEING DONE:

1) "$[\![C]\!]_q$" MEANS...? (AND IS IT SOUND?)

$\{(\langle \bar{a}_i, \bar{w}\rangle)\circ(\langle \bar{b}_i, \bar{w}\rangle) - \langle \bar{c}_i, \bar{w}\rangle \equiv 0 \pmod q\}_i$

$\mathbb{Z}$-R1CS: $\{(\langle \bar{a}_i, \bar{w}\rangle)\circ(\langle \bar{b}_i, \bar{w}\rangle) - \langle \bar{c}_i, \bar{w}\rangle = 0\}_i$

SPARSE VECTOR OF INTEGERS

- IF $C(\bar{w})=0$ THEN $[\![C]\!]_q(\bar{w}) \equiv 0 \pmod q$ ✓

AN ATTEMPT TO A TEMPLATE

$C(\bar{w})=0$

LET C BE SOME "COMPUTATION" OVER $\mathbb{Z}$.

PROVER $(\bar{w} \in \mathbb{Z}^m)$ | **VERIFIER** (vk_C)

$cm \leftarrow Com(\bar{w})$

$\xrightarrow{\quad cm \quad}$

$q \xleftarrow{\$} \mathbb{P}_\lambda$

$\xleftarrow{\quad q \quad}$

$[\![C]\!]_q :=$ "$C \bmod q$"

$\pi \leftarrow$ SOME SNARK.PROVE$\big(q,$ "$\exists\, \bar{w}$:
- $[\![C]\!]_q(\bar{w} \bmod q) = 0$
- $\bar{w}$ IS COMMITTED IN cm" $\big)$

$\xrightarrow{\quad \pi \quad}$

$Vfy_q\big(q, [\![vk]\!]_q, cm, \pi\big)$

BUT, WE ARE FAR FROM BEING DONE:

1) "$[\![C]\!]_q$" MEANS...? (AND IS IT SOUND?)

$$\Big\{\underbrace{(\langle \bar{a}_i, \bar{w}\rangle)\circ(\langle \bar{b}_i, \bar{w}\rangle)}_{\text{degree } 2} - \langle \bar{c}_i, \bar{w}\rangle \equiv 0 \pmod q\Big\}_i$$

Z-R1CS: $\Big\{(\langle \bar{a}_i, \bar{w}\rangle)\circ(\langle \bar{b}_i, \bar{w}\rangle) - \langle \bar{c}_i, \bar{w}\rangle = 0\Big\}_i$

SPARSE VECTOR OF INTEGERS

- IF $C(\bar{w})=0$ THEN $[\![C]\!]_q(\bar{w}) \equiv 0 \pmod q$ ✓
- IF $C(\bar{w}) \neq 0$ $\Pr_q\big[[\![C]\!]_q(\bar{w}) \equiv 0 \pmod q\big]$ IS LOW. ✓

JUST NOW:

FLAVOR OF THE SOLUTION

\+

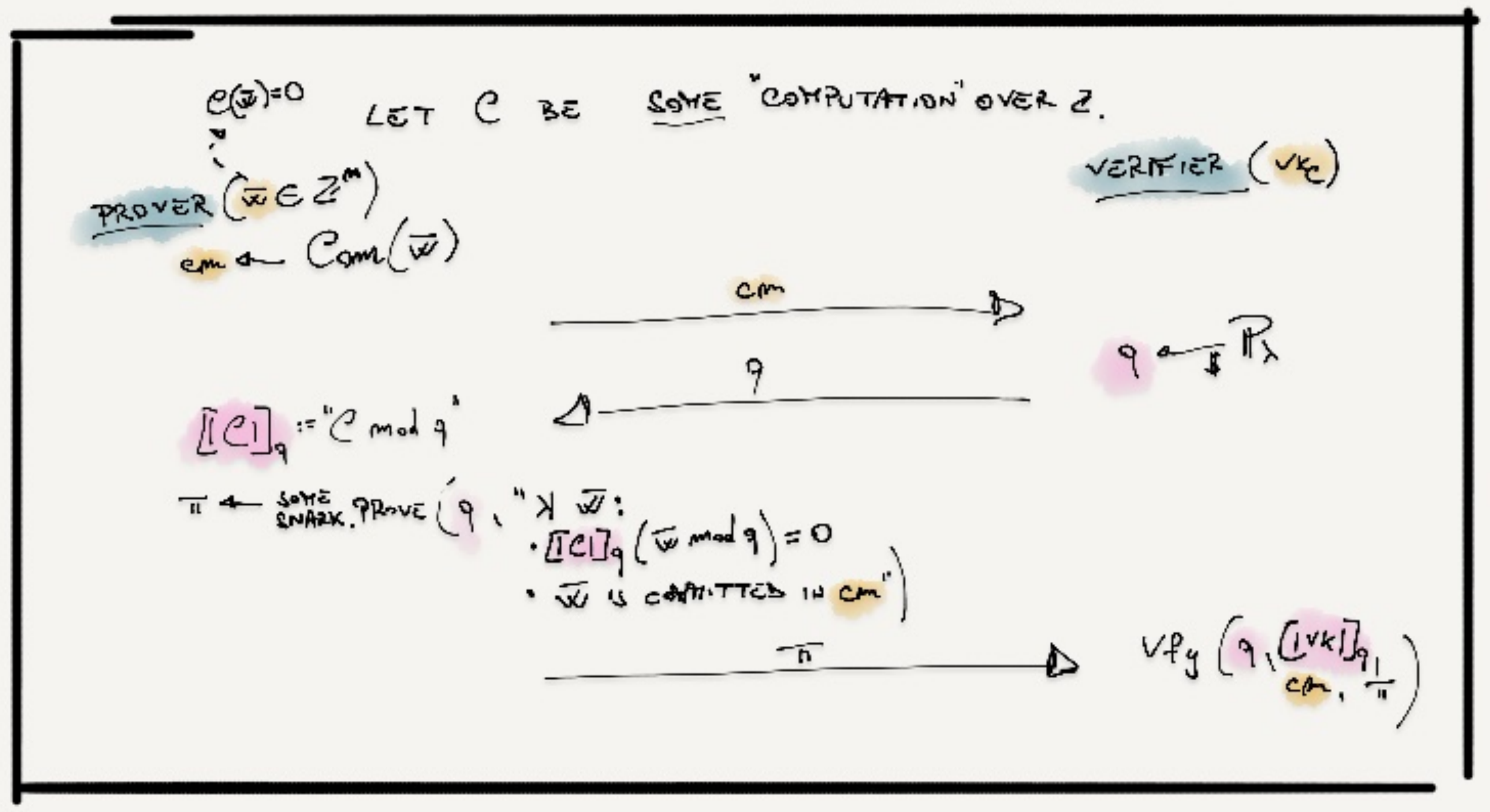
CHALLENGES

NEXT:

THE ACTUAL FRAMEWORK

↓

(SPOILER: VARIANT OF AHP + PC)



IMPLICITLY CAPTURED HERE

BRUSH UP ON AHPs/PIOPs/PHPs*

[MARLIN, DARK, PLONK, LUNAR, ...]

$\text{PROVER}^{\mathbb{F}_q}(w)$ — $\text{VERIFIER}^{\mathbb{F}_q}$

$p_1 \in \mathbb{F}_q[X, \ldots] \longrightarrow$

$\longleftarrow \alpha$

$p_2 \in \mathbb{F}_q[X, \ldots] \longrightarrow$

// HAS ORACLE ACCESS TO p_1, p_2

SAMPLE $\rho, \sigma, \ldots \leftarrow_\$ \mathbb{F}_q$

ASSERT:

$$p_1(\rho, \ldots) + \sigma\, p_2(\rho, \ldots) \stackrel{?}{=} \ldots$$

$$p_1(\sigma, \ldots) \cdot p_2(\sigma, \ldots) \stackrel{?}{=} \ldots$$

* ALGEBRAIC HOLOGRAPHIC PROOFS
POLYNOMIAL IOPs
POLYNOMIAL HOLOGRAPHIC IOPs

COMPILING AHPs,... VIA POLY COMMITMENTS

[MARLIN, DARK, PLONK, LUNAR, ...]

PROVER$^{\mathbb{F}_q}(w)$ — VERIFIER$^{\mathbb{F}_q}$

POLYNOMIAL COMMITMENT

$c_1 = \mathrm{Com}\left(p_1 \in \mathbb{F}_q[X,\ldots]\right) \longrightarrow$

$\longleftarrow \alpha$

$c_2 = \mathrm{Com}\left(p_2 \in \mathbb{F}_q[X,\ldots]\right) \longrightarrow$

// HAS ~~ORACLE ACCESS~~ COMMITMENTS TO p_1, p_2

SAMPLE $\rho, \sigma, \ldots \xleftarrow{\$} \mathbb{F}_q$

ASSERT: $a_1 + \sigma a_2 \stackrel{?}{=} \ldots$

$\cancel{p_1(\rho\ldots) + \sigma p_2(\rho\ldots) \stackrel{?}{=}} \ldots$

$\cancel{p_1(\sigma,\ldots)\cdot p_2(\sigma,\ldots) \stackrel{?}{=}} \ldots$

$b_1 \cdot b_2 \stackrel{?}{=} \ldots$

$a_1, a_2, b_1, b_2, \ldots \longrightarrow$

$\pi_1^{(a)}, \pi_2^{(a)}, \pi_1^{(b)}, \pi_2^{(b)} \longrightarrow$

PolyComm.Verify$\left(c_1, (\rho_1, \ldots, a_1, \pi_1^{(a)})\right)$

// ... CHECK OTHER PROOFS.

mod-AHPs: AHPs over $\mathbb{Z}$ w/ modular remainder queries

Prover ($\cancel{\mathbb{F}_q}$ w) — vector of integers

Verifier $\cancel{\mathbb{F}_q}$

$$\text{Prover} \xrightarrow{z \in \mathbb{Z}[X_1, \dots]} \text{Verifier}$$

$$\text{Prover} \xleftarrow{q} \text{Verifier} \qquad q \leftarrow_{\$} \mathbb{P}_\lambda$$

$$\text{Prover} \xrightarrow{p_1 \in \mathbb{F}_q[X_1, \dots]} \text{Verifier}$$

$$\text{Prover} \xleftarrow{\alpha} \text{Verifier}$$

$$\text{Prover} \xrightarrow{p_2 \in \mathbb{F}_q[X_1, \dots]} \text{Verifier}$$

// has oracle access to z, p_1, p_2 (mod q)

Sample $\rho_1, \sigma, \dots \leftarrow_{\$} \mathbb{F}_q$

Assert:

$$z(\rho_1 \dots) \cdot p_1(\rho_1 \dots) + \sigma p_2(\rho_1 \dots) \stackrel{?}{=} \dots \pmod{q}$$

$$p_1(\sigma_1 \dots) \cdot p_2(\sigma_1, \dots) \stackrel{?}{=} \dots \pmod{q}$$

mod-AHPs: AHPs over $\mathbb{Z}$ w/ modular remainder queries

PROVER (w) $\cancel{\mathbb{F}_q}$ → VECTOR OF INTEGERS

VERIFIER $\cancel{\mathbb{F}_q}$

$z \in \mathbb{Z}[X_1, \dots]$ →

$q \leftarrow_{\$} \mathbb{P}_\lambda$

← q

$p_1 \in \mathbb{F}_q[X_1, \dots]$ →

← α

$p_2 \in \mathbb{F}_q[X_1, \dots]$ →

// HAS ORACLE ACCESS TO z, p_1, p_2 $\underline{(\text{mod } q)}$

SAMPLE $\rho, \sigma, \dots \leftarrow_{\$} \mathbb{F}_q$

ASSERT:

$$z(\rho_1 \dots) \cdot p_1(\rho_1 \dots) + \sigma\, p_2(\rho_1 \dots) \stackrel{?}{=} \dots \pmod q$$

$$p_1(\sigma_1 \dots) \cdot p_2(\sigma_1 \dots) \stackrel{?}{=} \dots \pmod q$$

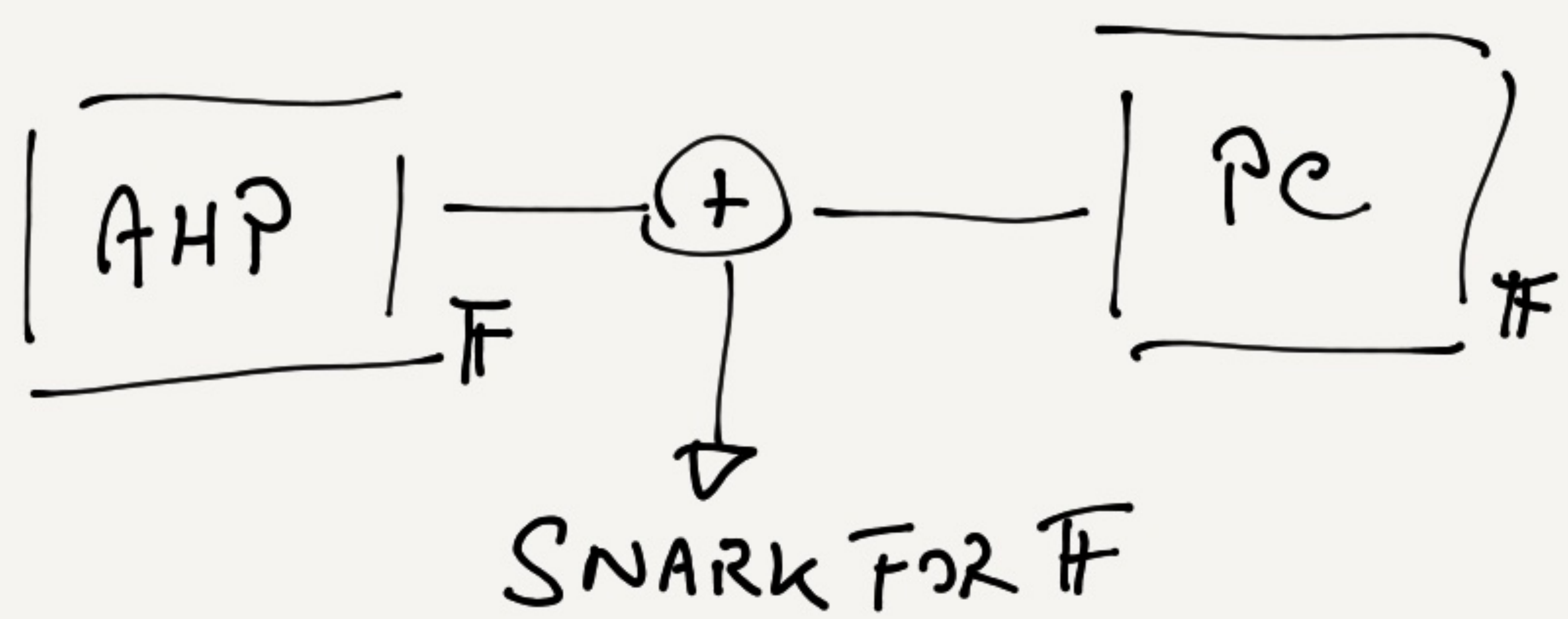
NB: WE NEVER EVALUATE ANYTHING OVER $\mathbb{Z}$ HERE

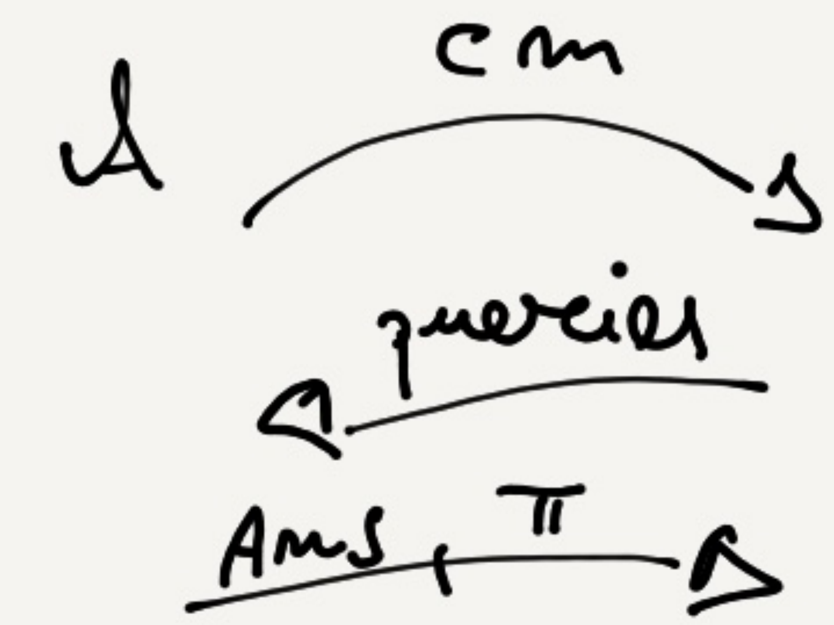
⇓

FIRST MILESTONE TOWARDS FULL-SUCCINCTNESS.

AHP + PC $\longmapsto$ mod-AHP + ?

+ Poly Comm





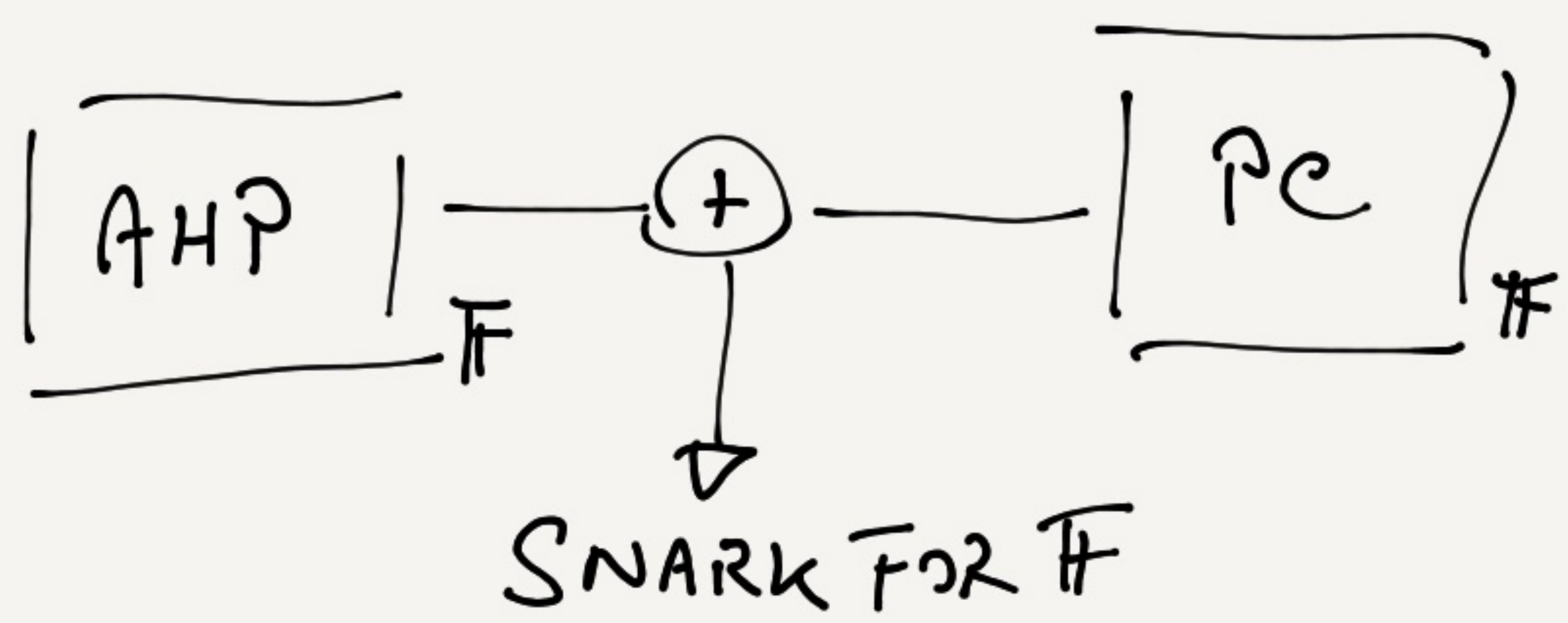
Ext $\to g$:

$\mathrm{opn}(cm) = g$

$g(\text{queries}) = \text{Ans}$

AHP + PC ⟼ mod-AHP + q
+ Poly Comm

STANDARD CASE



cm

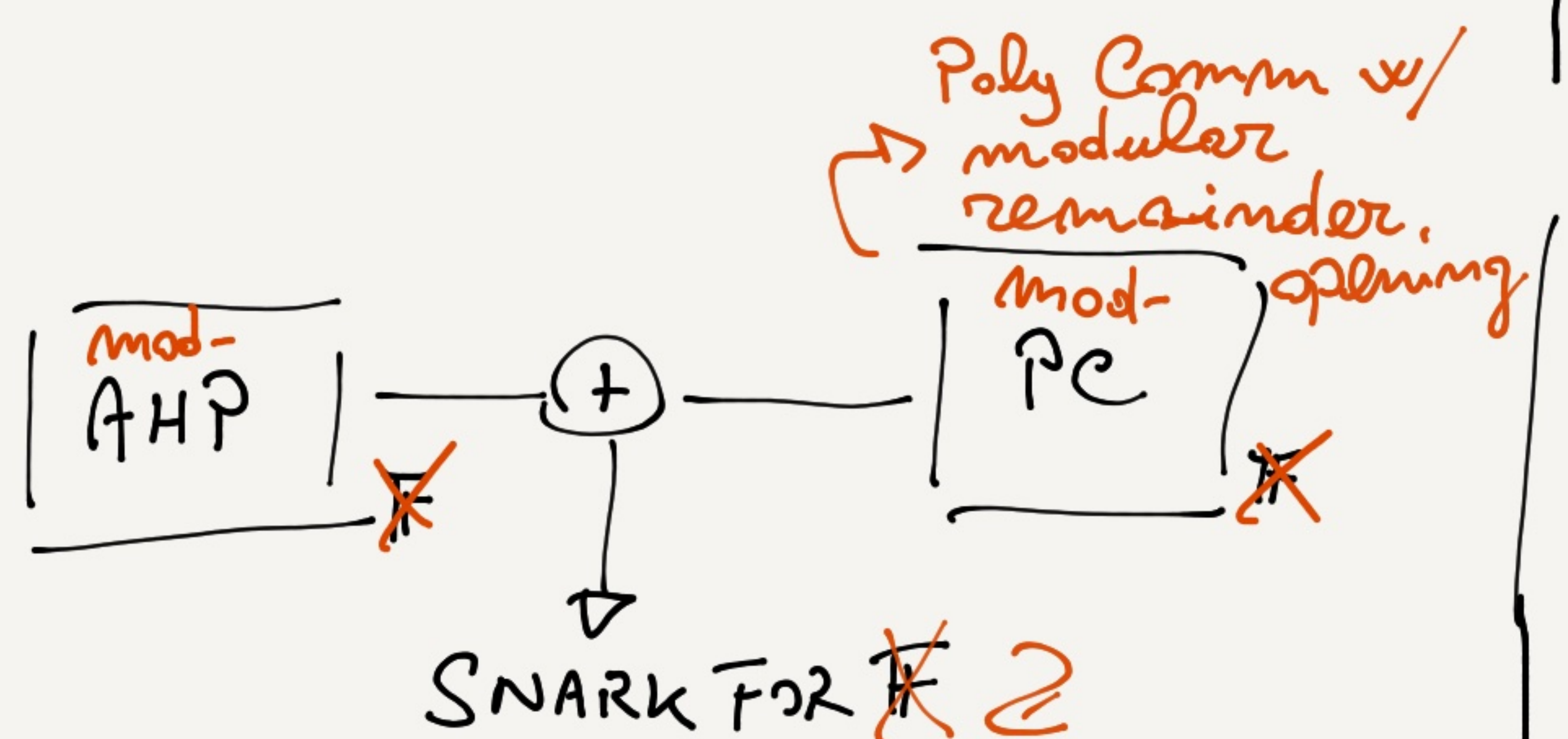
queries

Ans, π

Ext → g :

opn($\hat{cm}$) = g

g (queries) = Ans



cm

q

queries

Ans, π

$q \leftarrow^{\$} P_\lambda$

Ext → g :

opn($\hat{cm}$) = g

g (queries) ≡ Ans (mod q)

KSND in mod-AHP

KSND AHP (STANDARD) : $Ext \to \overline{w} \in \mathbb{F}^n$:

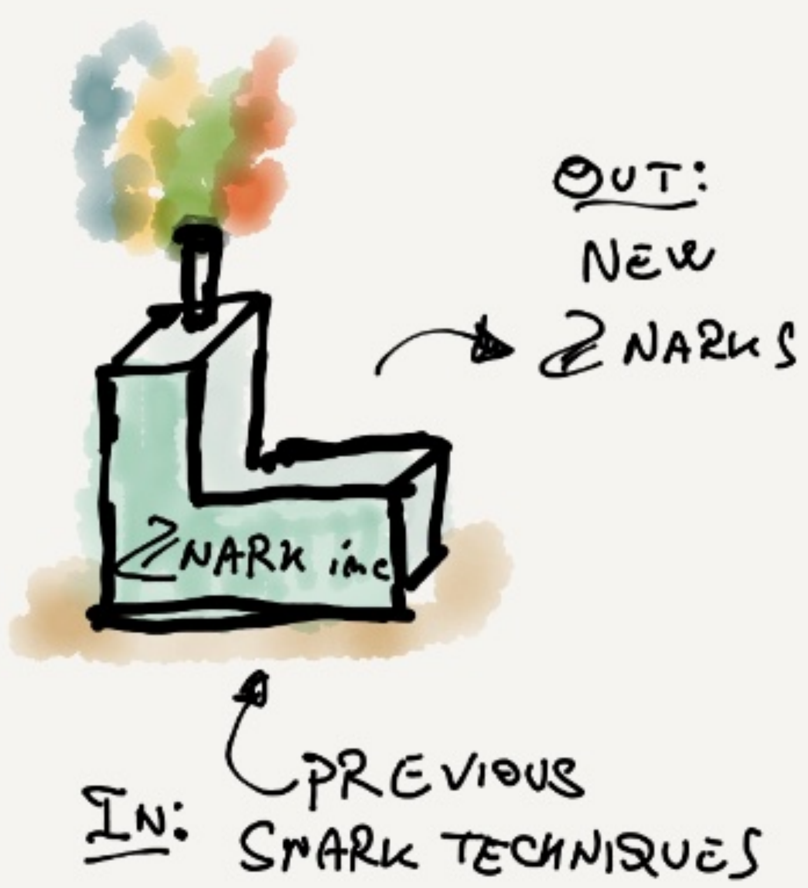
$C(\overline{w})$ ✓

NATURAL KSND mod-AHP : $Ext \to \overline{w} \in \mathbb{Z}^n$

$C(\overline{w})$ ✓

KSND IN mod-AHP

KSND AHP (STANDARD) : Ext $\to \bar{w} \in \mathbb{F}^n$:
$C(\bar{w})$ ✓



NATURAL KSND mod-AHP : Ext $\to \bar{w} \in \mathbb{Z}^n$
$C(\bar{w})$ ✓

IMPLIES

SIMPLER ("WEAK") KSND mod-AHP : Ext $\to \bar{w} \in \mathbb{F}_q^n$ $(q \leftarrow_\$ \mathbb{P}_\lambda)$
$[C]_q(\bar{w})$ ✓

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$\boxed{A}\,\bar{w} \circ \boxed{B}\,\bar{w} - \boxed{C}\,\bar{w} = 0$

VARIANT OF SPARTAN (SETTY '20)

$P(\bar{w})$ $\quad$ $V^{\hat{A},\hat{B},\hat{C}}$

$\tilde{w}(\bar{X}) = \mathrm{MLE}(\bar{w}) \longrightarrow$

$q \leftarrow \mathbb{P}_\lambda$

OVER $\mathbb{F}_q$: $\boxed{\text{SUMCHECK 1}}$ … $\boxed{\text{SUMCHECK 2}}$

$\ldots + \tilde{w}(\rho) + \ldots \overset{?}{\equiv} 0 \pmod{q}$

$\tilde{A}(\sigma, \tau) \ldots \tilde{B}(\sigma, \tau) \ldots \equiv 0 \pmod{q}$

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$\boxed{A}\,\bar{w}\circ\boxed{B}\,\bar{w}-\boxed{C}\,\bar{w}=0$

VARIANT OF SPARTAN (SETTY '20)

$P(\bar{w})$ $\qquad\qquad$ $V^{\tilde{A},\tilde{B},\tilde{C}}$

$\tilde{w}(\bar{X})=\mathrm{MLE}(\bar{w}) \longrightarrow$

$q \leftarrow \mathbb{P}_\lambda$

OVER $\mathbb{F}_q$:

SUMCHECK1

...

SUMCHECK2

$\dots+\tilde{w}(\rho)+\dots \stackrel{?}{\equiv} 0 \pmod q$

$\tilde{A}(\sigma,\tau)\dots\tilde{B}(\sigma,\tau)\dots\equiv 0 \pmod q$

<u>EASY</u>: PROVING

SIMPLER ("WEAK") KSND mod-AHP: $Ext \to \bar{w}\in\mathbb{F}_q^m$ $(q \leftarrow_\$ \mathbb{P}_\lambda)$ $[\![C]\!]_q(\bar{w})$ ✓

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$\boxed{A}\,\bar{w} \circ \boxed{B}\,\bar{w} - \boxed{C}\,\bar{w} = 0$

VARIANT OF SPARTAN (SETTY '20)

$P(\bar{w})$ $\qquad V^{\hat{A},\hat{B},\hat{C}}$

$\tilde{w}(\bar{X}) = \text{MLE}(\bar{w})$

$q \leftarrow \mathbb{P}_\lambda$

OVER $\mathbb{F}_q$

SUMCHECK1

...

SUMCHECK2

$\ldots + \tilde{w}(\rho) + \ldots \stackrel{?}{\equiv} 0 \pmod q$

$\tilde{A}(\sigma, \tau) \ldots \tilde{B}(\sigma, \tau) \ldots \equiv 0 \pmod q$

$\tilde{A}$, etc. HAVE QUADRATIC SIZE BUT ARE SPARSE.

EASY: PROVING

SIMPLER ("WEAK") KSND mod-AHP: $\text{Ext} \to \bar{w} \in \mathbb{F}_q^m$ $(q \leftarrow_\$ \mathbb{P}_\lambda)$ $[\![C]\!]_q(\bar{w})$ ✓

NOT EASY: MAKING V NOT PAY FOR $\tilde{A}, \tilde{B}, \tilde{C}$

SPARTAN'S DENSE 2 SPARSE COMPILER $\leftarrow$ EXTENSIONS TO FRAMEWORK ("DELAYED-INPUT" mod-AHPs, STRONGER mod-PC)

LENS: TECHNIQUES/RECIPES FOR "mod-FUNCTIONAL COMMITMENTS"

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$\boxed{A}\bar{w} \circ \boxed{B}\bar{w} - \boxed{C}\bar{w} = 0$

VARIANT OF SPARTAN (SETTY '20)

$P(\bar{w})$ $\qquad$ $V^{\tilde{A},\tilde{B},\tilde{C}}$

$\tilde{w}(\bar{X}) = \mathrm{MLE}(\bar{w})$

$q \leftarrow \mathbb{P}_\tau$

OVER $\mathbb{F}_q$:

SUMCHECK 1

...

SUMCHECK 2

$\dots + \tilde{w}(\rho) + \dots \stackrel{?}{\equiv} 0 \pmod{q}$

$\tilde{A}(\sigma,\tau) \dots \tilde{B}(\sigma,\tau) \dots \equiv 0 \pmod{q}$

EFFICIENCY FEATURES (FROM SPARTAN):

- LINEAR
- VERY PARALLELIZABLE (SIMD)

<u>EXAMPLE</u>: FOR RSA SIGNATURES ($|\bar{w}| = 16$) YOU ARE RUNNING SPARTAN ON A <u>SIZE 16 CIRCUIT</u> (3 ORDERS OF MAGNITUDE LESS THAN THE FIELD-EMULATION CASE)

INSTANTIATIONS (AND THEIR CHALLENGES): mod-PC

OUR STARTING POINT:

TECHNIQUES FROM GROUPS OF UNKNOWN ORDER

(DARK (EC '20), Block et al. (CRYPTO '21))

$\mathrm{Com}(f) = g^{f(Q)}$

Q: LARGE INTEGER

INSTANTIATIONS (AND THEIR CHALLENGES): mod-PC

OUR STARTING POINT: TECHNIQUES FROM GROUPS OF UNKNOWN ORDER (DARK (EC '20), Block et al. (CRYPTO '21))

$Com(f) = g^{f(Q)}$

Q: LARGE INTEGER

CHALLENGE 1: BINDING OF DARK INSUFFICIENT

DOES NOT RECOVER $f(\bar{X})$ (A POLY)

BUT $\frac{f(\bar{X})}{N}$ (A RATIONAL FUNCTION)

INSTANTIATIONS (AND THEIR CHALLENGES): mod-PC

OUR STARTING POINT: TECHNIQUES FROM GROUPS OF UNKNOWN ORDER (DARK (EC '20), Block et al. (CRYPTO '21))

$\mathrm{Com}(f) = g^{f(Q)}$

Q: LARGE INTEGER

CHALLENGE 1: BINDING OF DARK INSUFFICIENT

USE Block et al. INSTEAD AS MAIN TOOL

DOES NOT RECOVER $f(\bar{X})$ (A POLY)

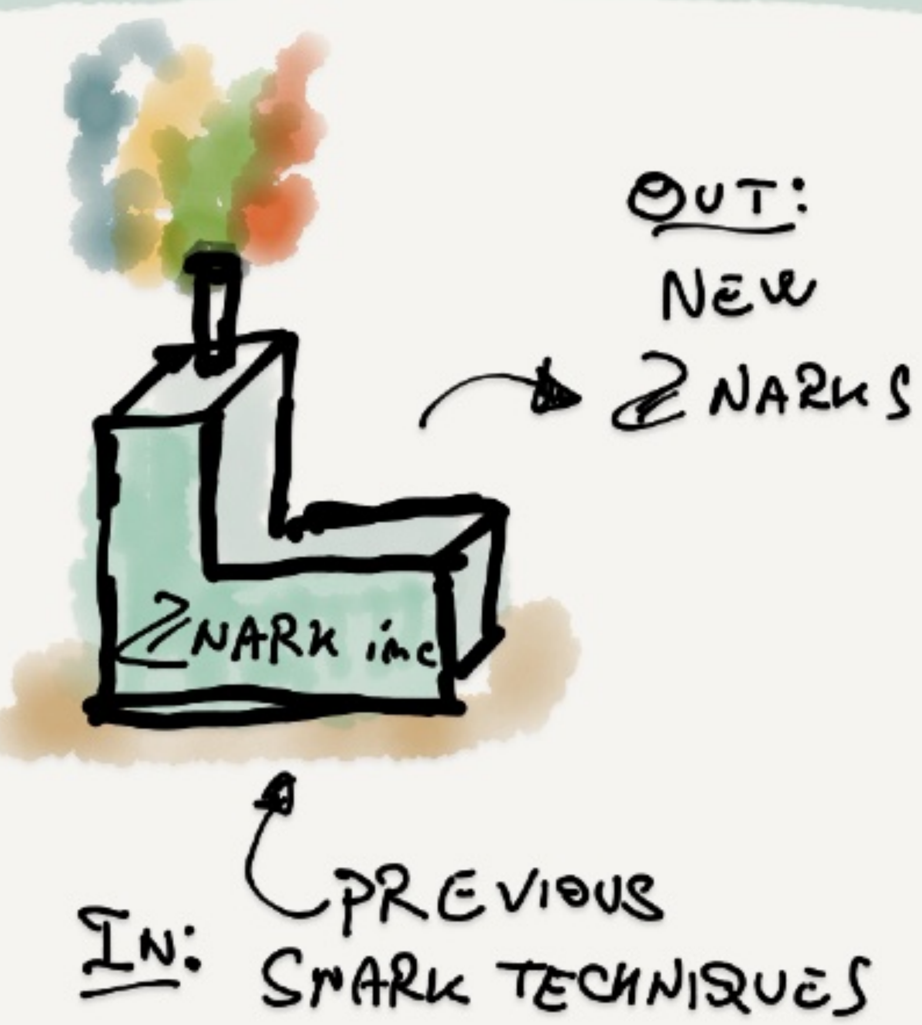
BUT $\frac{f(\bar{X})}{N}$ (A RATIONAL FUNCTION)

CHALLENGE 2: NOT SUCCINCT IN $\|\bar{w}\|_\infty$

REQUIRE ADDITIONAL BUILDING BLOCKS (APPROPRIATE ARG OF KNOWLEDGE)

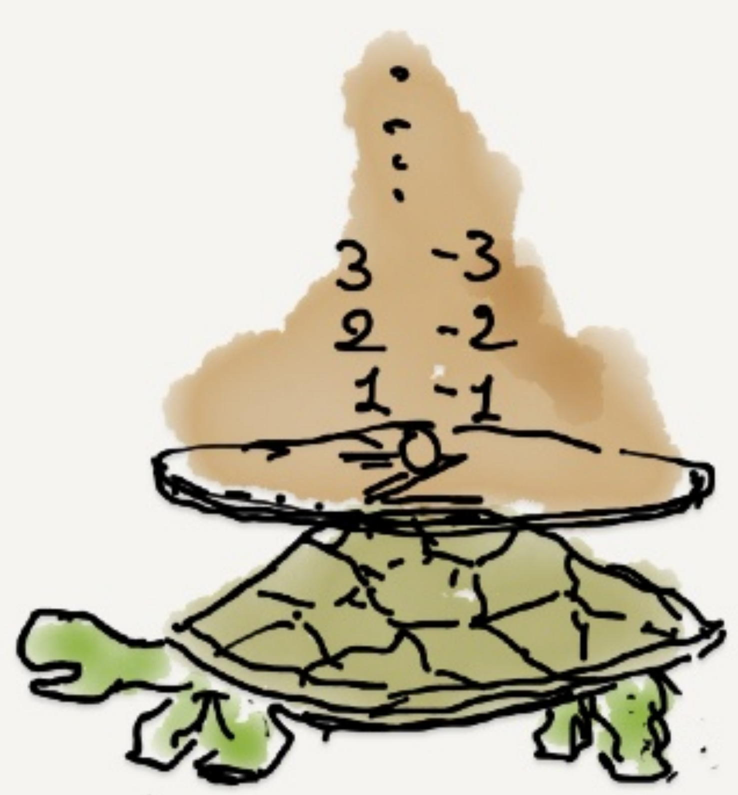
WRAPPING UP

TECHNIQUES & FRAMEWORK FOR (FULLY SUCCINCT) ZNARKS



BUILDING BLOCKS/INSTANTIATIONS 

CONCRETE CONSTRUCTION : ZARATAN (SPARTAN FOR Z)



FUTURE WORK:

- ZK
- DIFFERENT mod-PCs AND TECHNIQUES
- MORE mod-AHP INSTANTIATIONS (HyperPlonk, WHAT ELSE?)

THANKS!