

ZNARKs

SNARKs For INTEGER COMPUTATIONS

MATTEO
CAMPANELLI*
(OFFCHAIN LABS)

&

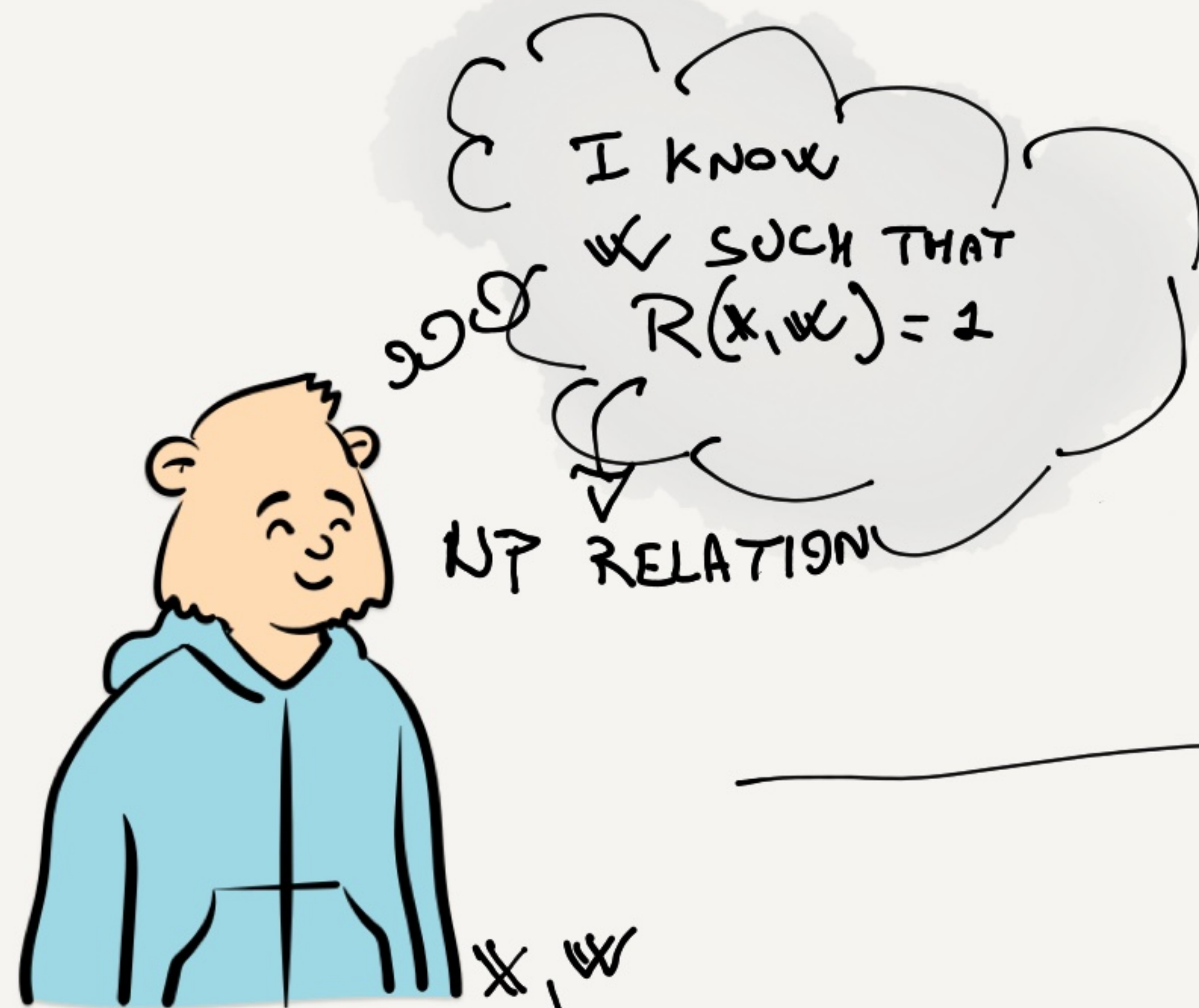
MATHIAS
HALL-ANDERSEN**
(ZKSECURITY)

"FULLY SUCCINCT ARGUMENTS
OVER THE INTEGERS
FROM FIRST PRINCIPLES"

* BINARY WHALES.COM
** ROT256.DEV

EPRINT : 2024/1548

SNARK BRUSHUP



NP RELATION

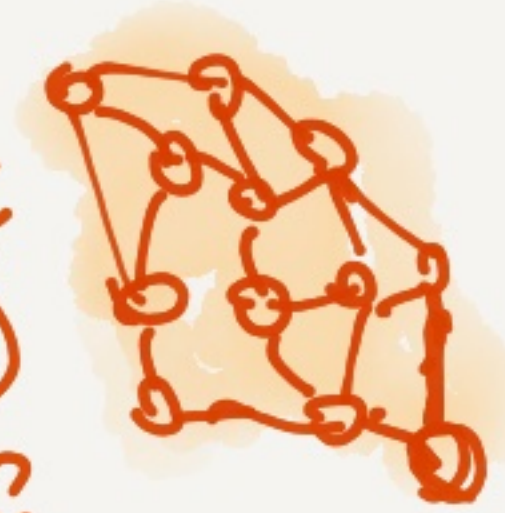
π



x, w
 $R(x, w)$

e.g.: SOME HASH
 h

- ITS PREIMAGE (A GRAPH G)
- A 3-COLORING FOR G



SECURITY:

EXTRACTABILITY

SUCCINCTNESS:

$$|\pi| \ll |w|$$

THE PRESENT: EFFICIENT SNARKS

FOR MANY COMPUTATIONS

CONCRETELY

• A CLASSICAL
EXAMPLE

GROTH16

THE PRESENT: EFFICIENT SNARKS

FOR MANY COMPUTATIONS

CONCRETELY

- A CLASSICAL EXAMPLE

GROTH16

- BUT THAT IS NOT THE END OF THE LINE.

VIRGO

LIGERO

DARK

VAMPIR

HYRAX

DEK

SPARTAN

LUNAR

BASILISK

ECLIPSE

WHIR/STEER/FRI*

BULLETPROOF

PLONK

BASEFOLD

ORION

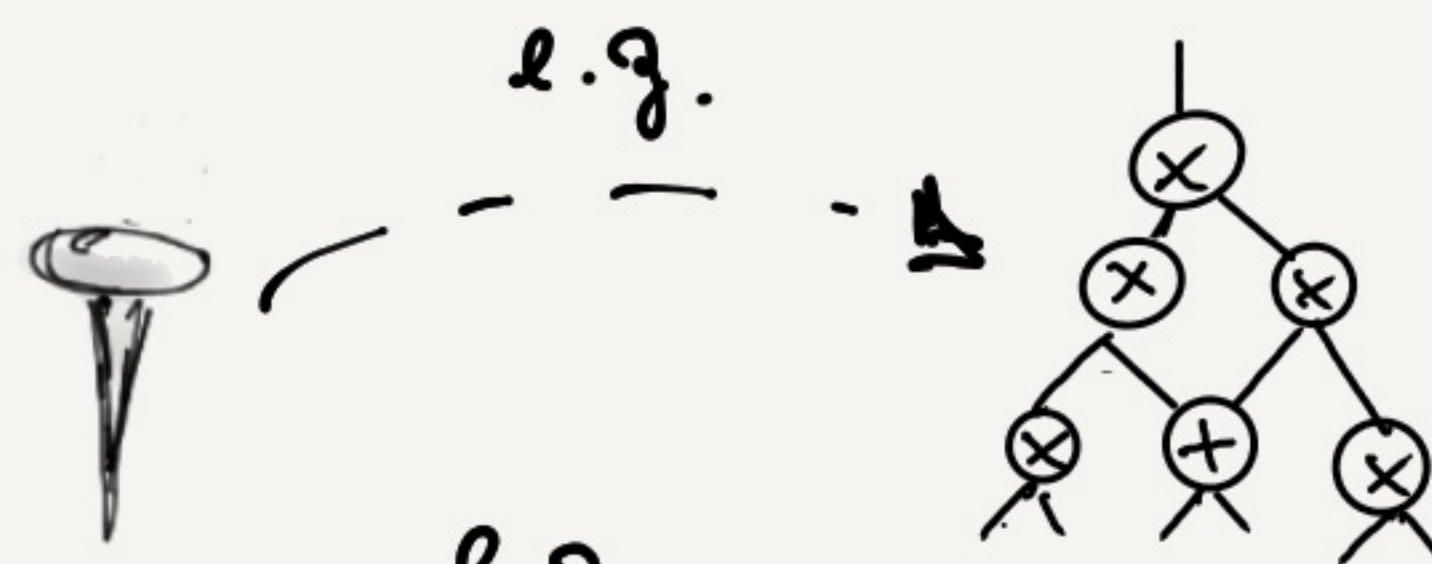
NOVA

SOLT

(AND MORE...)

CAVEAT: GOOD EFFICIENCY

RELIES ON "GOOD REPRESENTATION"

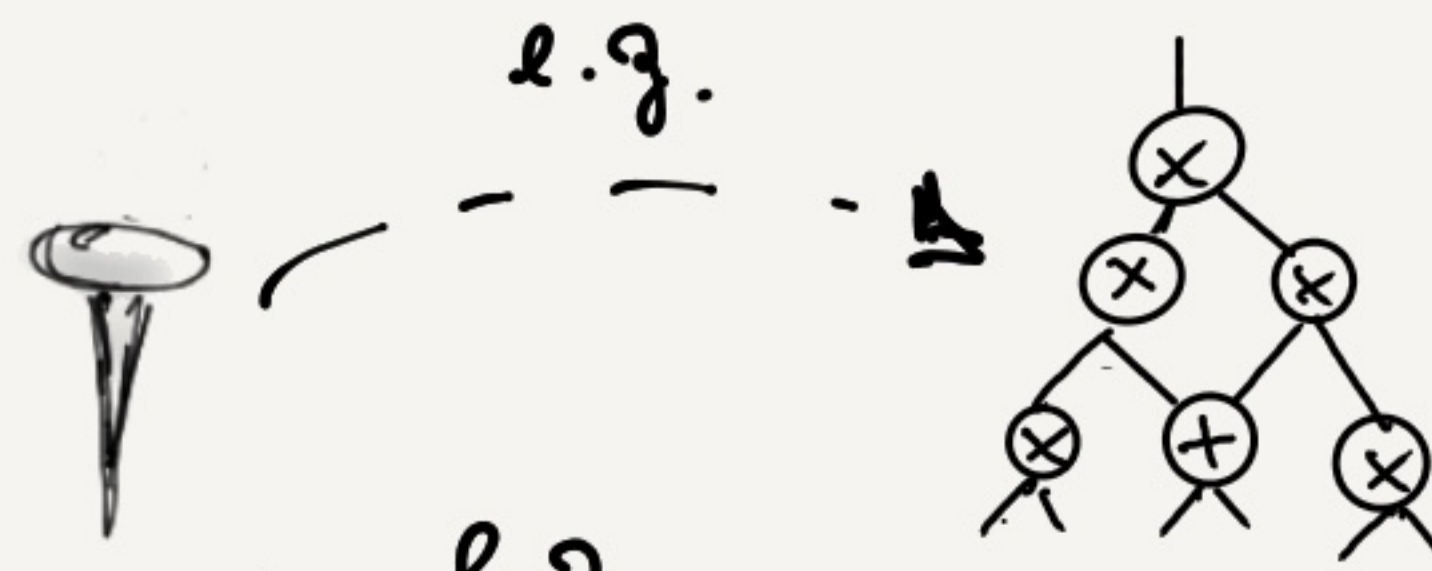
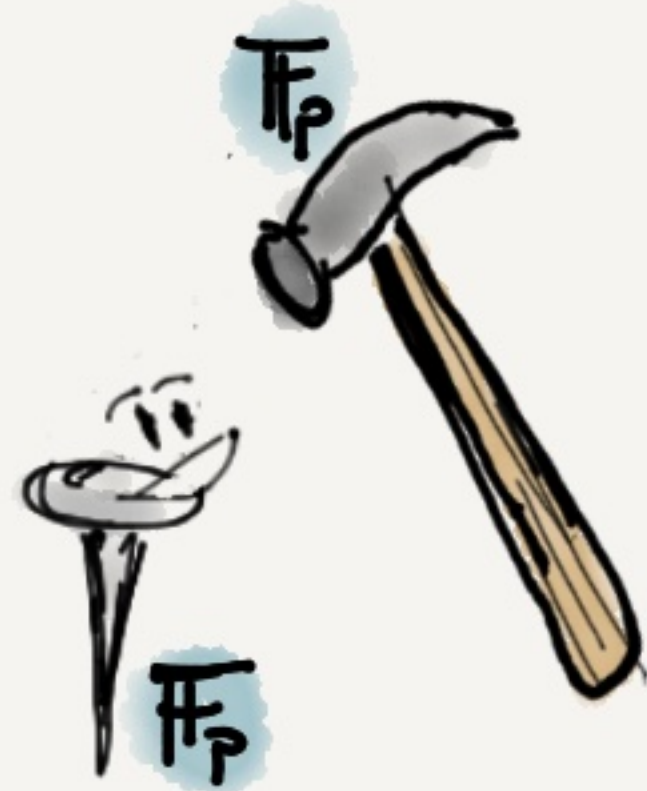


e.g.

$$\left(\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) - \left(\begin{bmatrix} B \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) = \bar{0}$$

CAVEAT: GOOD EFFICIENCY

RELIES ON "GOOD REPRESENTATION"



$$\left(\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) \cdot \left(\begin{bmatrix} B \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) - \left(\begin{bmatrix} C \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) = \bar{0}$$

BEYOND FINITE FIELDS?

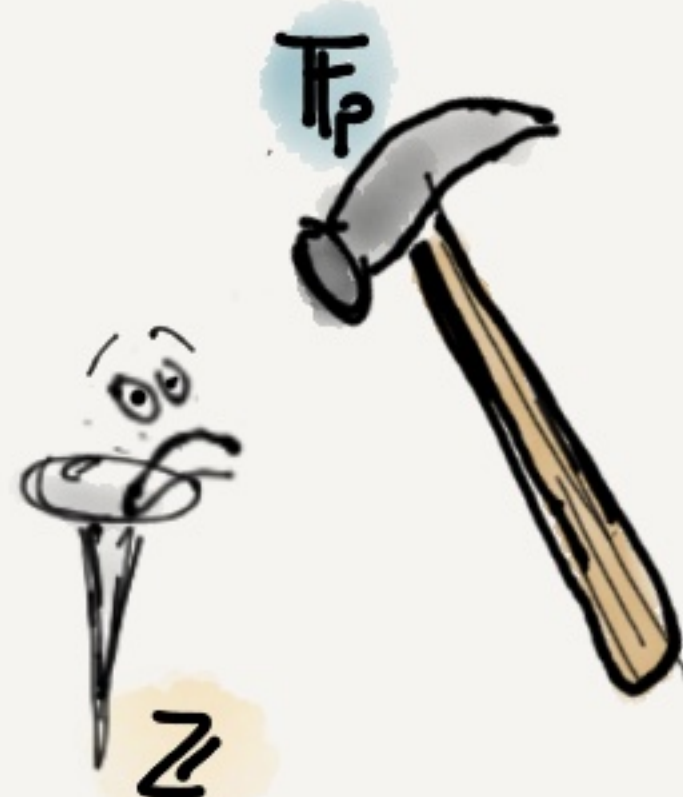
- NOT ALL COMPUTATIONS ADMIT "EASY" REPRESENTATIONS OVER FINITE FIELDS.

EASY EXAMPLE: RSA SIGNATURE VERIFICATION

$$\text{Verify}(pk, m, \sigma) := \sigma^2 \equiv m \pmod{N}$$

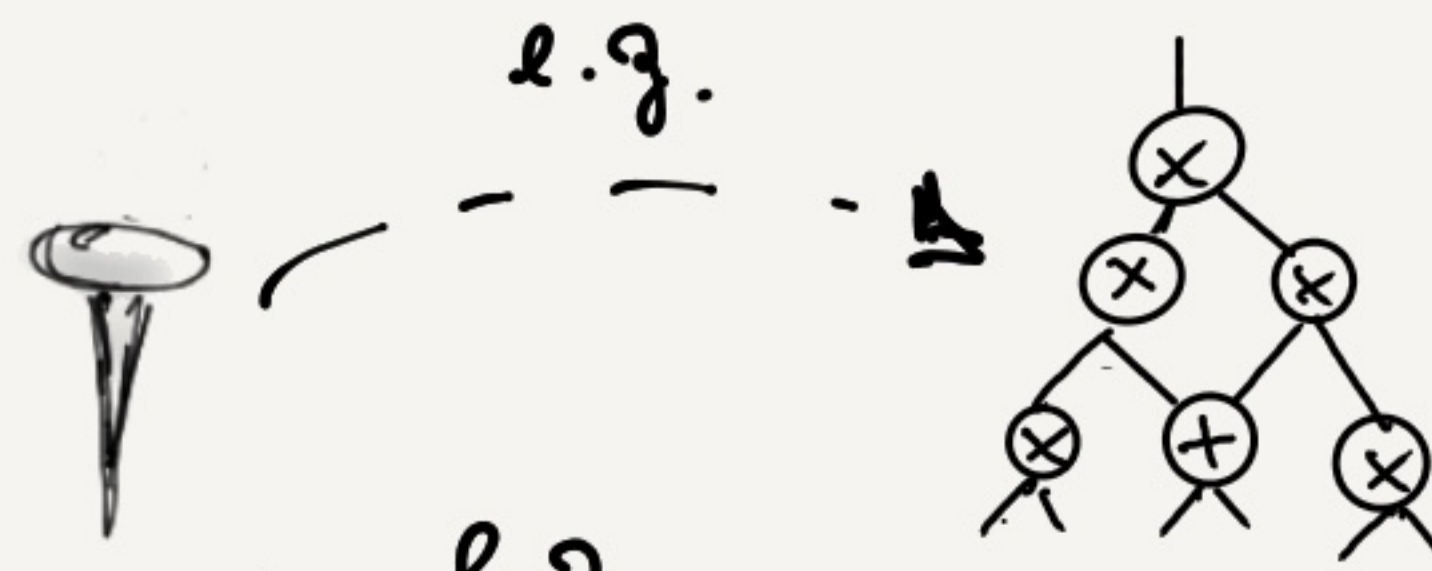
BUT ALSO:

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



CAVEAT: GOOD EFFICIENCY

RELIES ON "GOOD REPRESENTATION"



$$\left(\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) \cdot \left(\begin{bmatrix} B \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) - \left(\begin{bmatrix} C \end{bmatrix} \cdot \begin{bmatrix} z \end{bmatrix} \right) = \bar{0}$$

BEYOND FINITE FIELDS?

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

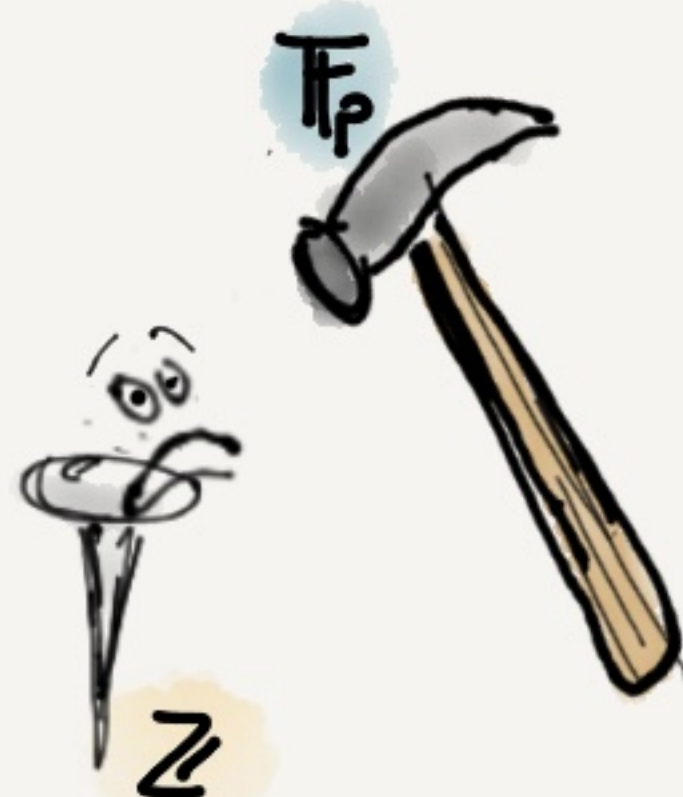
EASY EXAMPLE: RSA SIGNATURE VERIFICATION

$$\text{Verify}(\text{pk}, m, \sigma) := \sigma^2 \equiv m \pmod{N}$$

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



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



CASE STUDY: RSA SIGNATURES

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

SIGNATURE MESSAGE PK

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

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

$$qN + m = \sigma^e$$

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

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 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!"

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

SUCCINCT** (NON-INTERACTIVE)

\mathbb{Z} MARKS:

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

SUCCINCT** (NON-INTERACTIVE)

ZNARKS:

ARGUMENTS THAT WORK
NATIVELY* OVER \mathbb{Z}

⇒ CONCRETE CONSTRUCTIONS

⇒ GENERAL TECHNIQUES / FRAMEWORK

* "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)

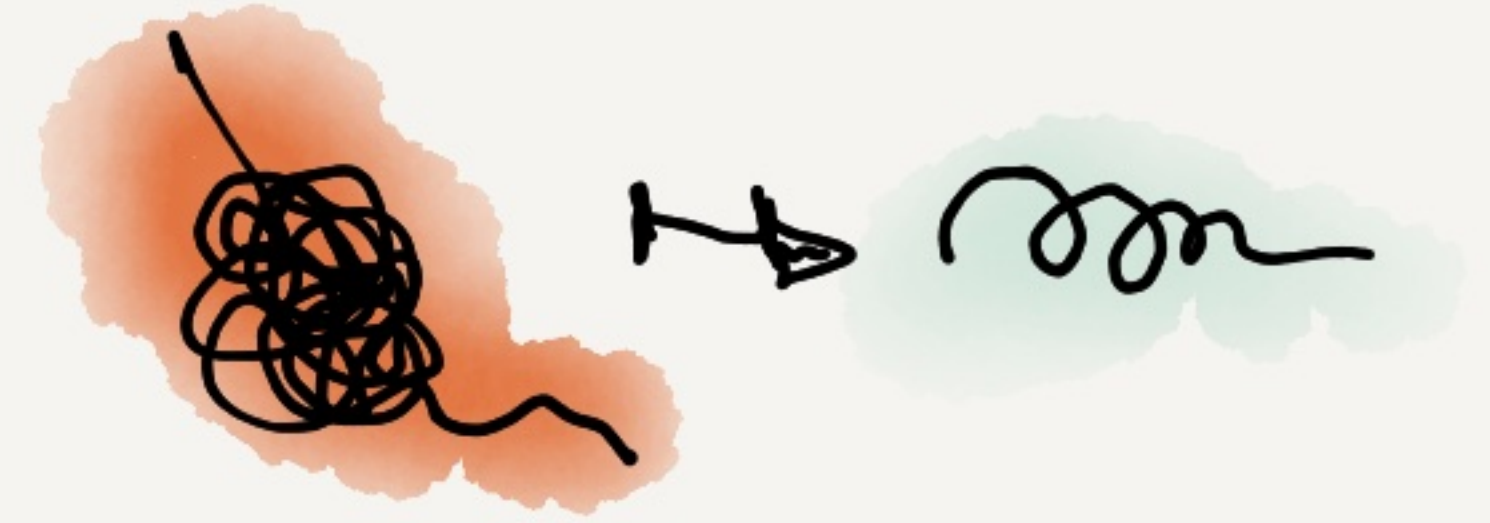
NEXT:

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 $x \in \mathbb{Z}$. $x \geq 0 \iff \exists \alpha, \beta, \gamma, \delta \in \mathbb{Z}$:
(SUM OF 4 SQUARES) $x = \alpha^2 + \beta^2 + \gamma^2 + \delta^2$

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

SNARKS FOR \mathbb{Q}

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

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

• ADD: (SIMILAR)

• τ , check $den \neq 0$: ADAPT TECHNIQUES FROM
PREV. SLIDE

ARBITRARY PRECISION ARITHMETIC

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

POTENTIAL APPLICATIONS:

SCIENTIFIC COMPUTATION,
VAT STATEMENTS,
(AND MORE)

NEW APPROACHES / AVENUES FOR SNARKS FOR TIL

TWO AVENUES

1) DIRECT:

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

2) INDIRECT:

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

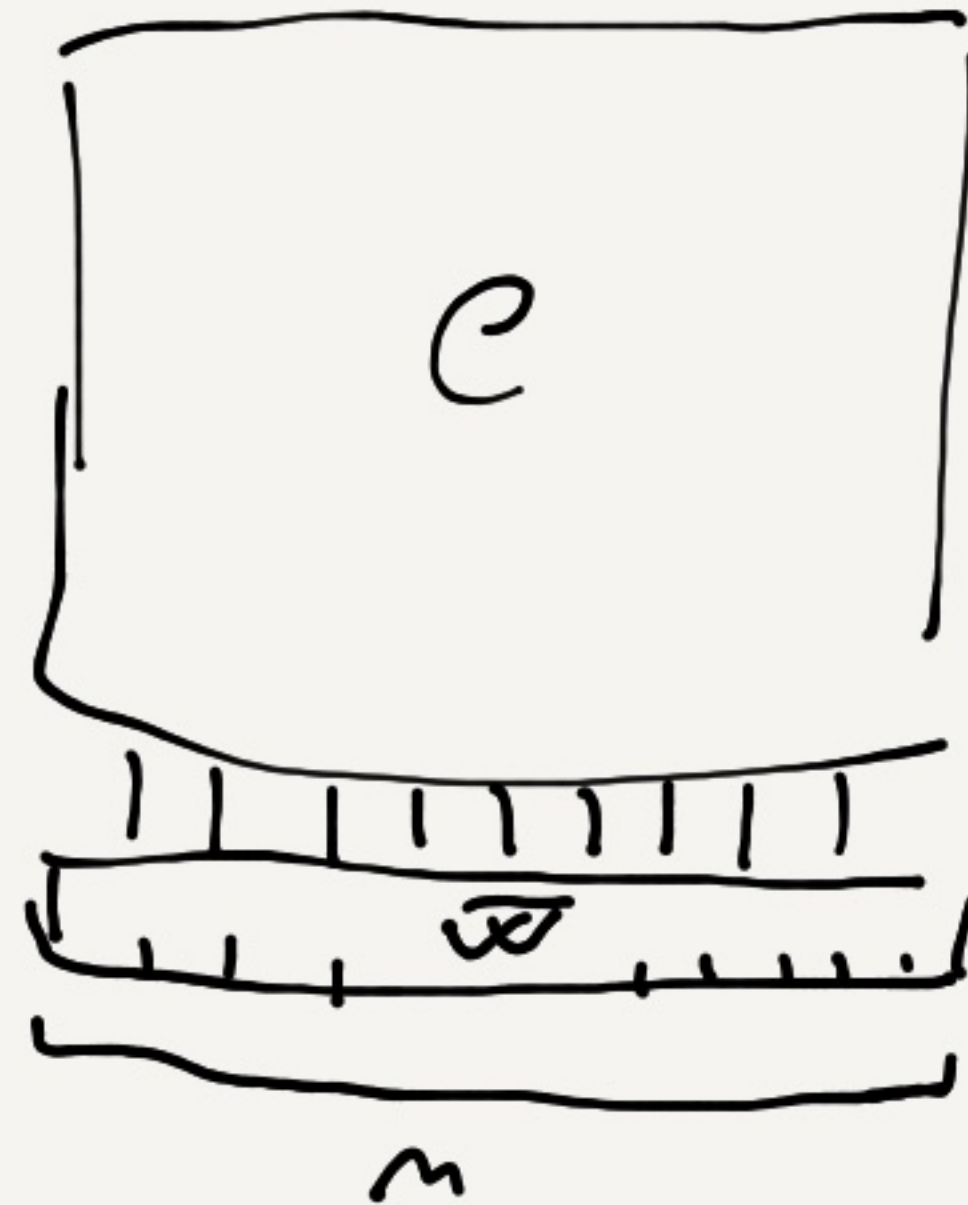
BUT ENOUGH
ABOUT MOTIVATION.
WHAT ABOUT THIS WORK?

(ALMOST THERE, BUT FIRST:

- A NOTE ON SUCCINCTNESS
- PRIOR WORK)

"Succinctness"

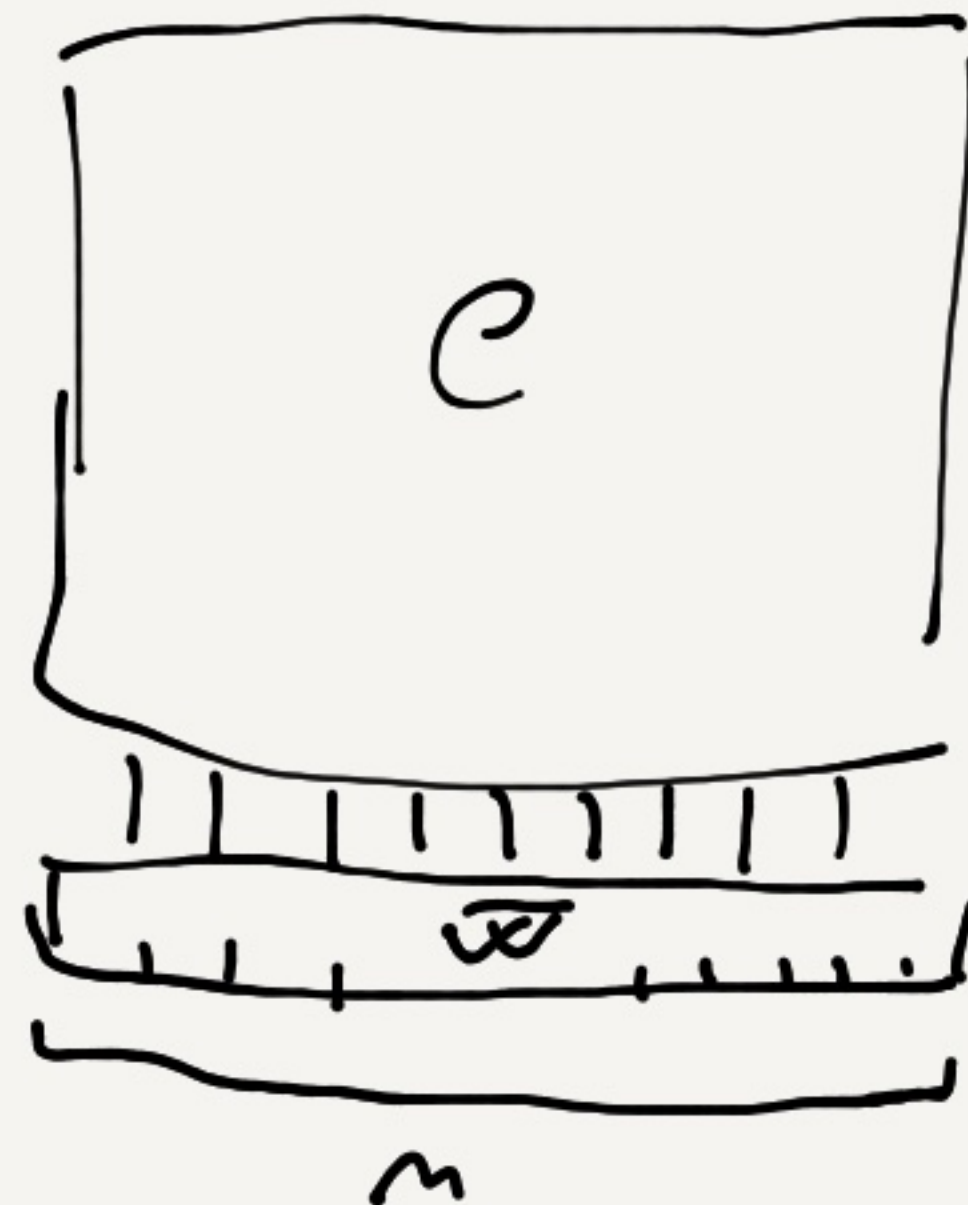
For Σ Computations



$$\text{succinctness (traditionally)} := \frac{|\pi| \in \text{TIME}(v)}{\text{small on}} \circ (|w|)$$

"SUCCINCTNESS"

FOR \mathbb{Z} COMPUTATIONS



$$\text{SUCCINCTNESS (TRADITIONALLY)} := \frac{|\pi| \in \text{TIME}(v) \in \text{O}(|w|)}{\text{SMALL ON}}$$

BUT, OVER \mathbb{Z} , WHAT ABOUT

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

$$|\bar{w}| = 3$$

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

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

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}

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 $\exists z_4$
(WE HAVE NO z_k)
/ Δ LEAKS $\|w\|_\infty$

PRIOR WORK

SNARKS OVER RINGS:

PINOCCHIO
(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 & VERGAUD, AC'20)

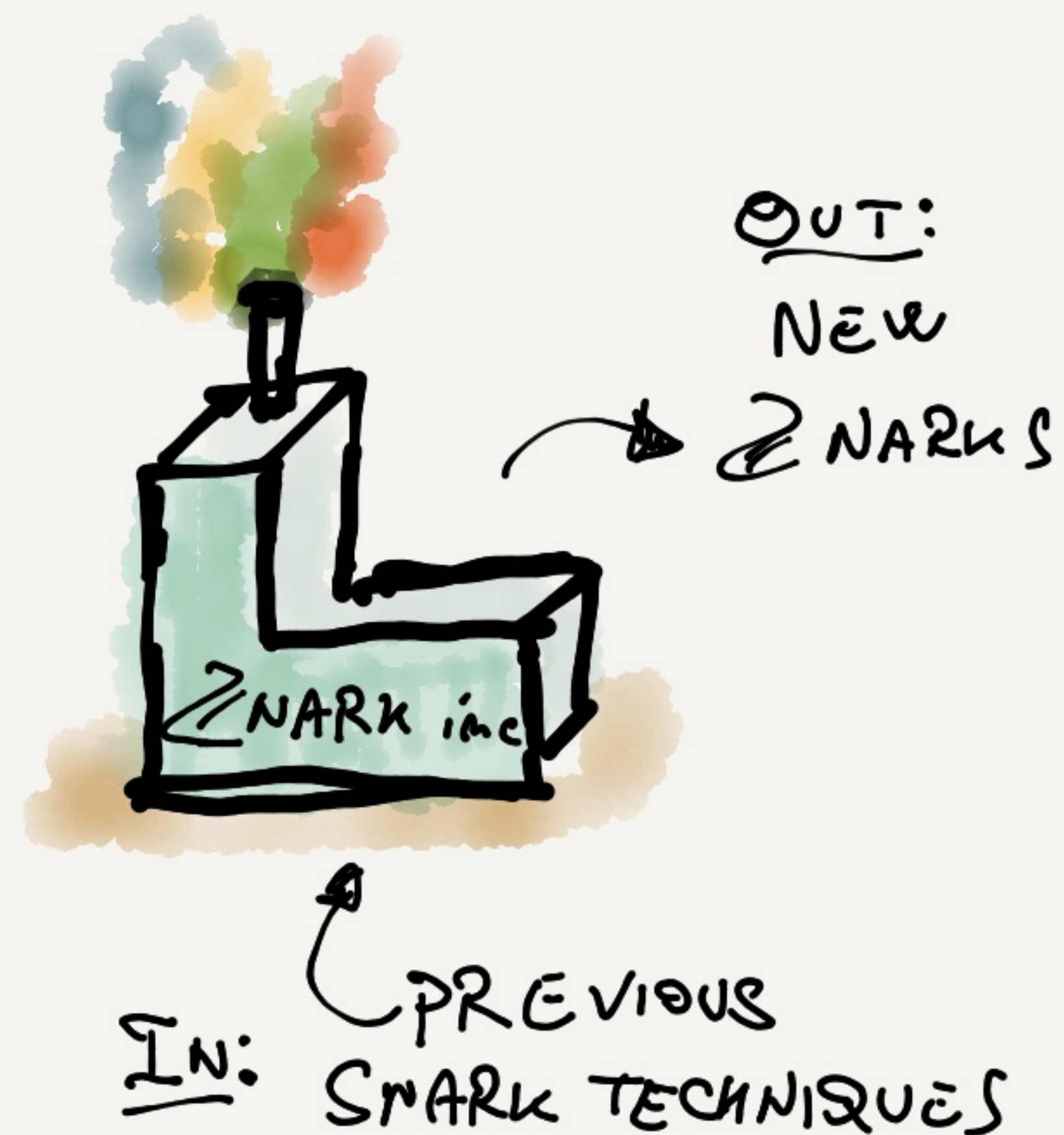
BP

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

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

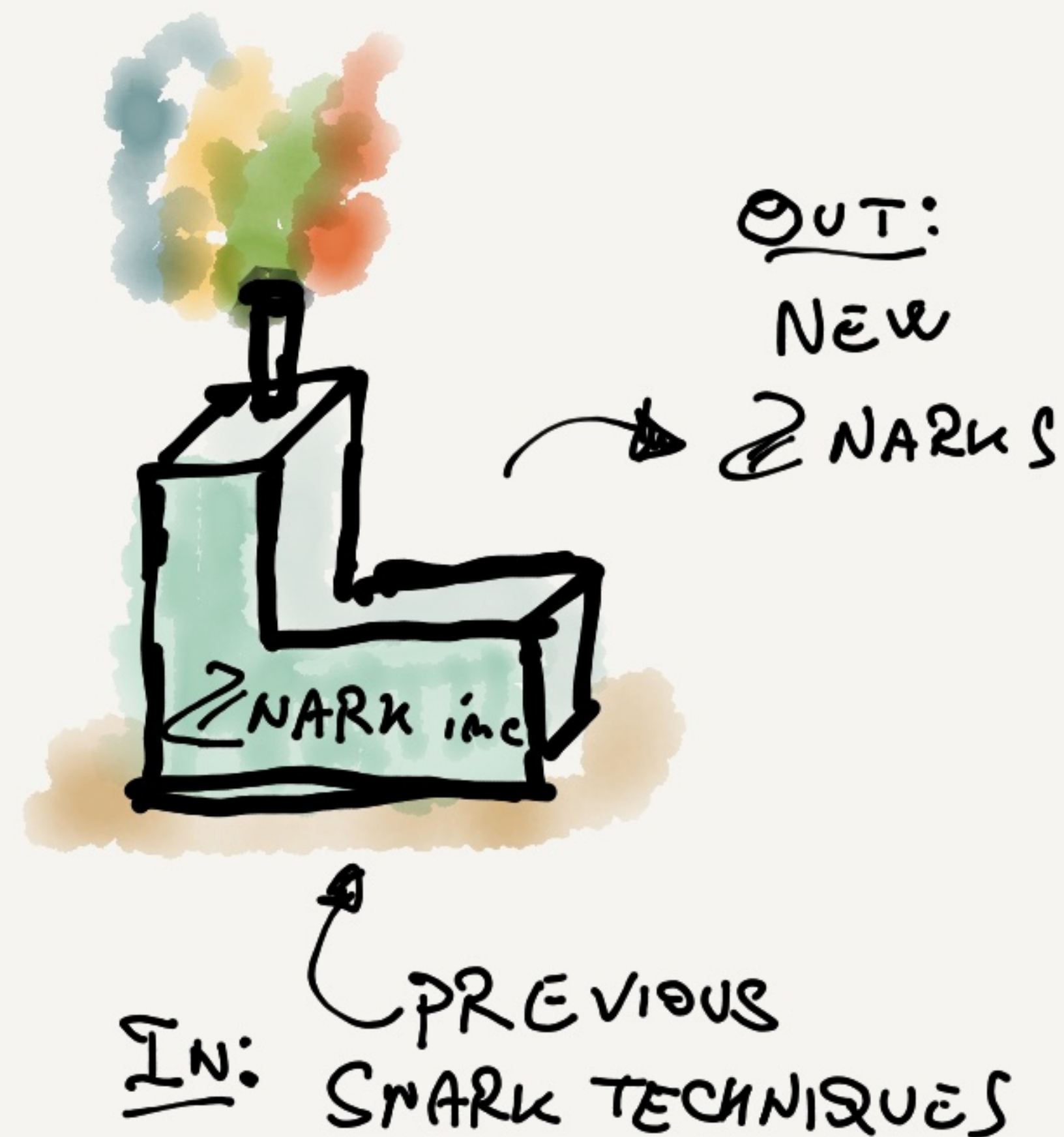
THIS WORK

TECHNIQUES
&
FRAMEWORK
FOR
(FULLY
SUCCINCT) ZNARKS



THIS WORK

TECHNIQUES
&
FRAMEWORK
FOR
(FULLY
SUCCINCT) ZNARKS

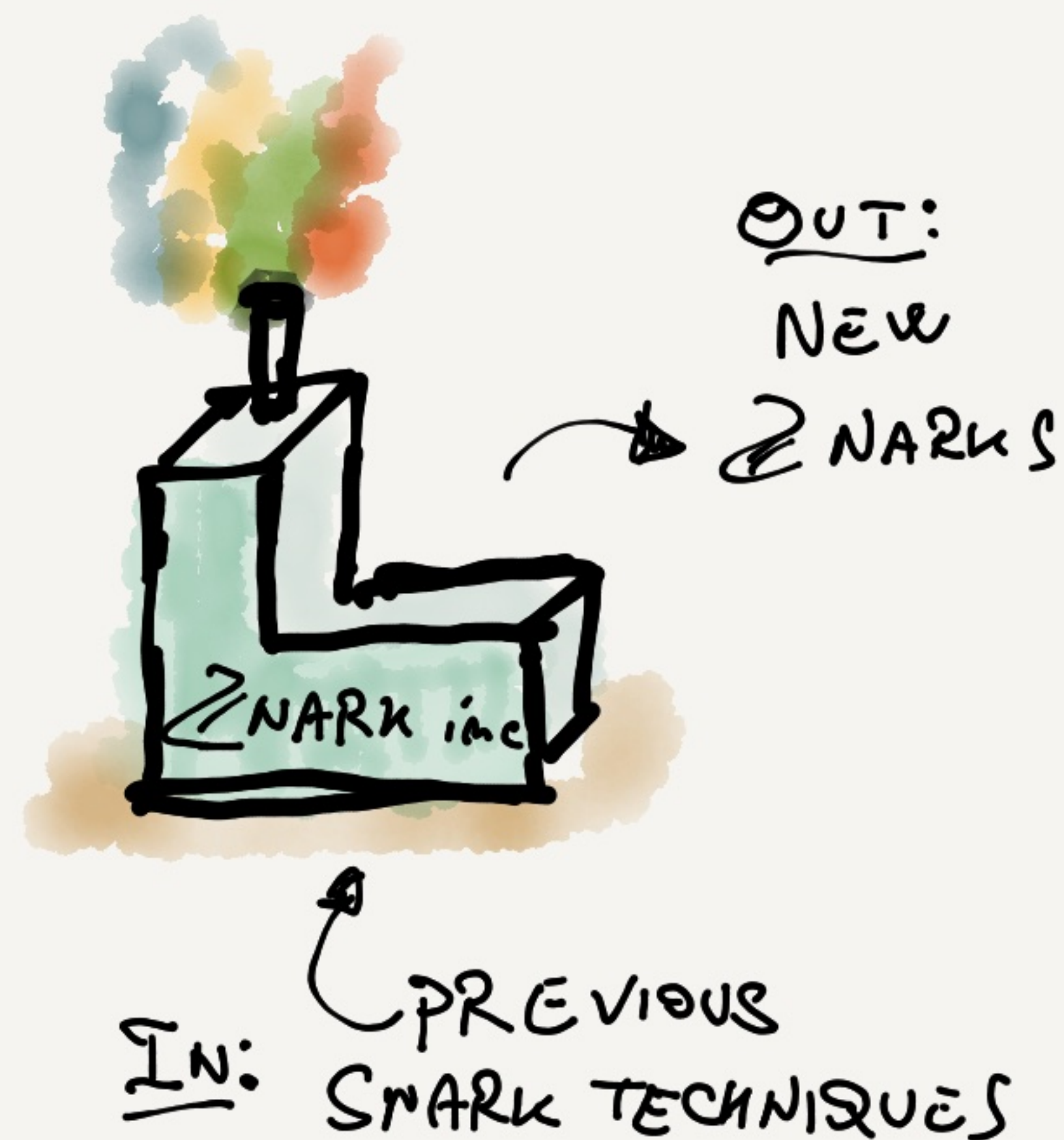


BUILDING
BLOCKS/INSTANTIATIONS



THIS WORK

TECHNIQUES
&
FRAMEWORK
FOR
(FULLY
SUCCINCT) ZNARKS

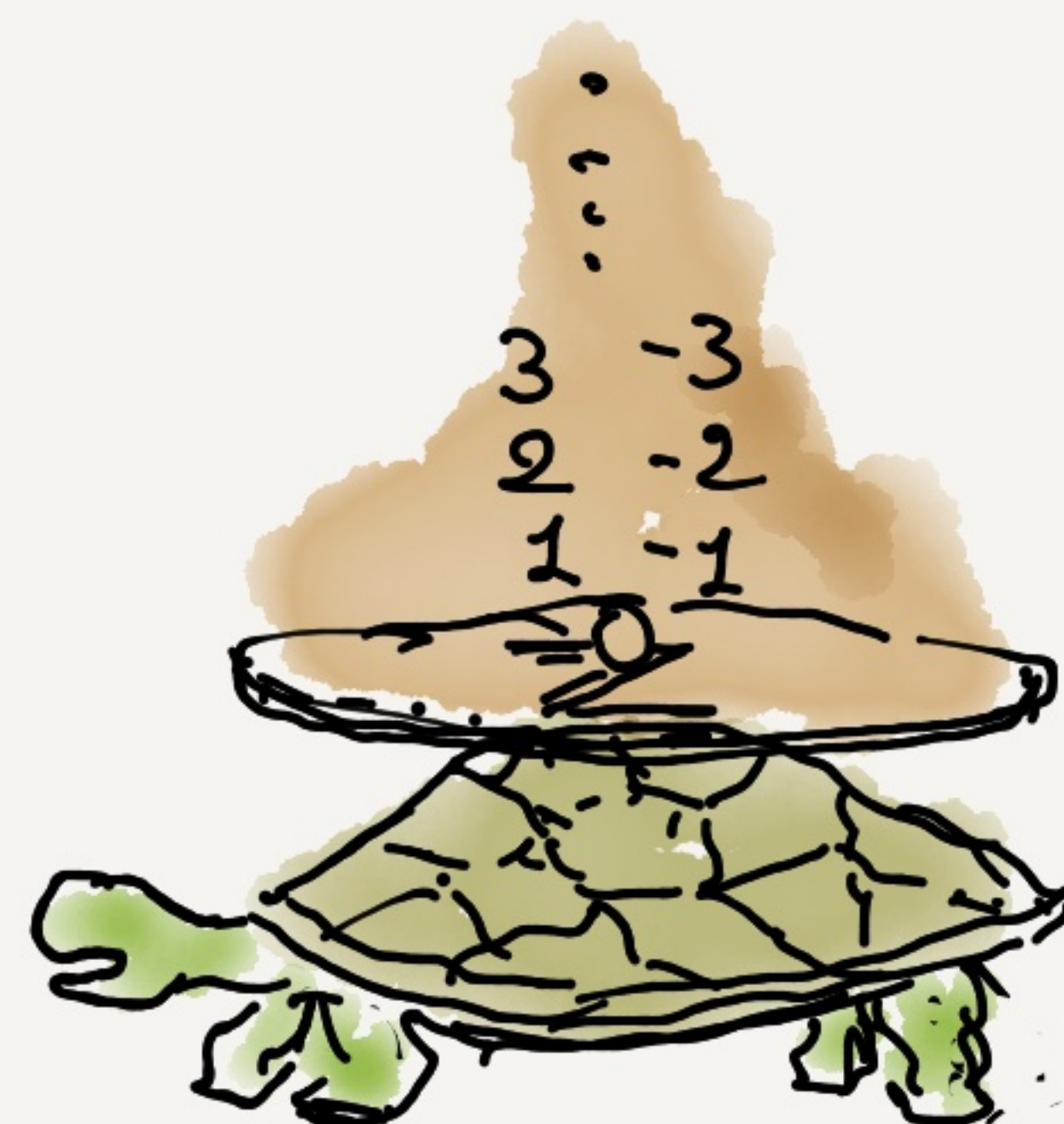


BUILDING
BLOCKS/INSTANTIATIONS



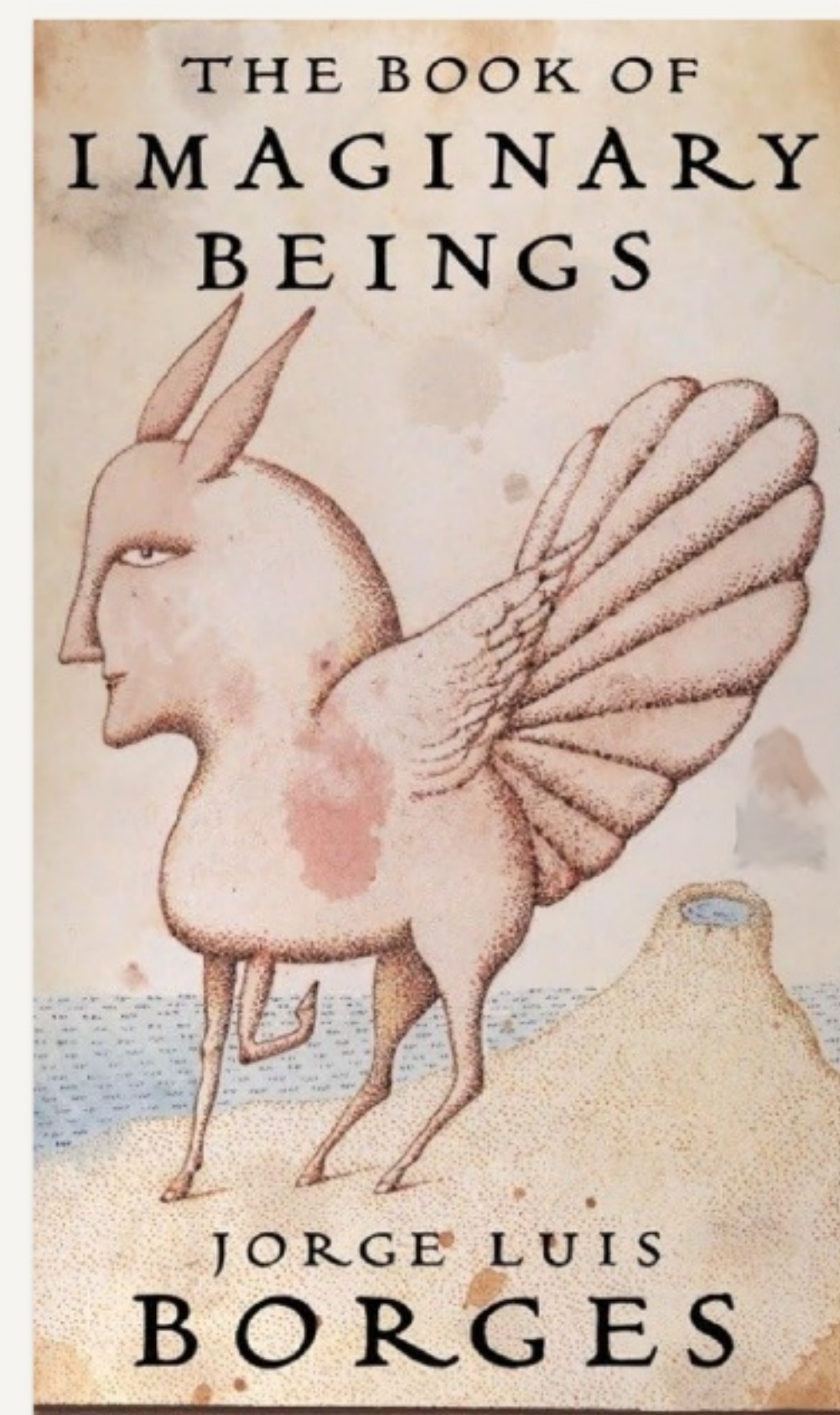
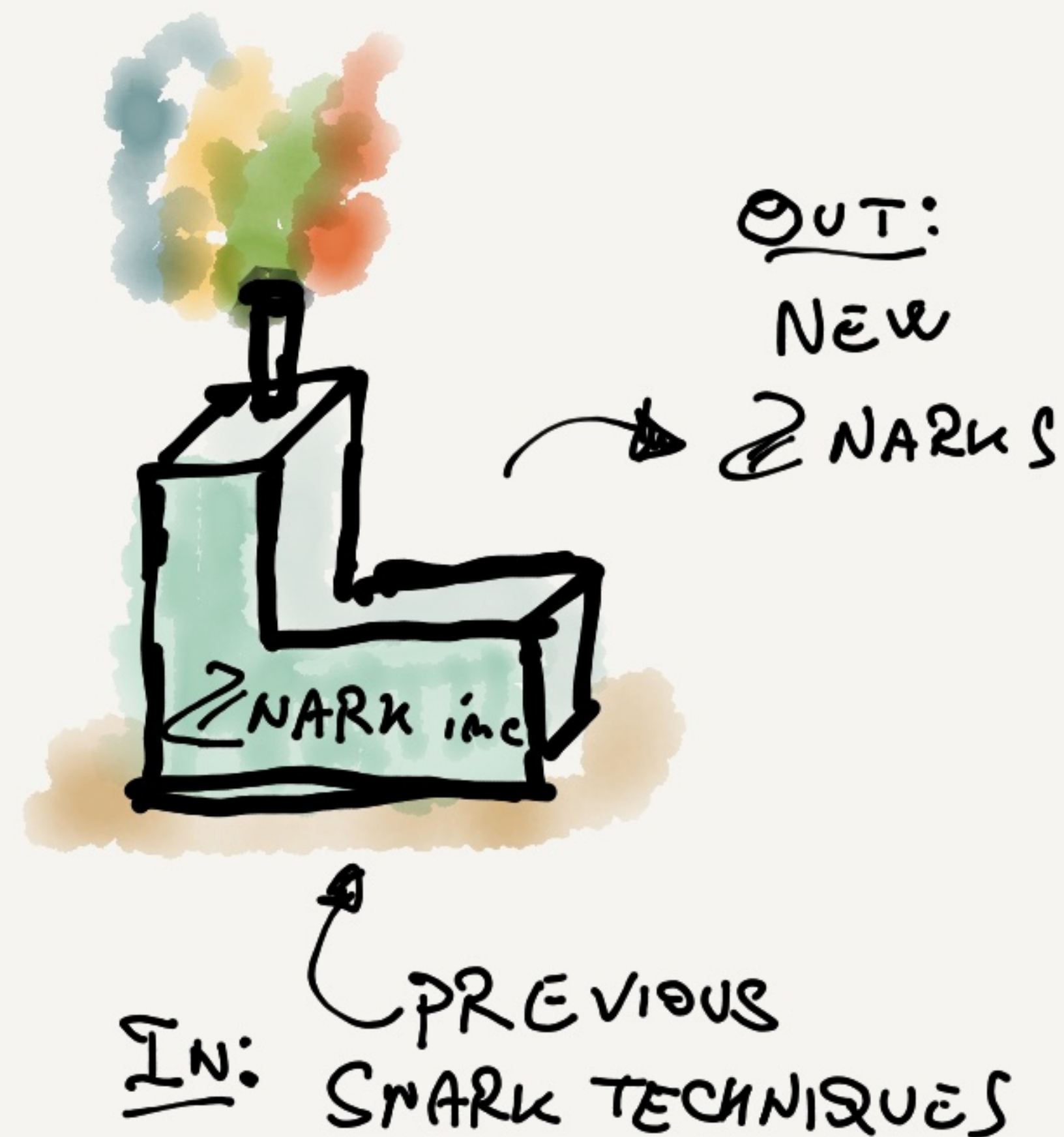
CONCRETE
CONSTRUCTION

: ZARATAN
(SPARTAN FOR Z)



THIS WORK

TECHNIQUES
&
FRAMEWORK
FOR
(FULLY
SUCCINCT) ZNARKS



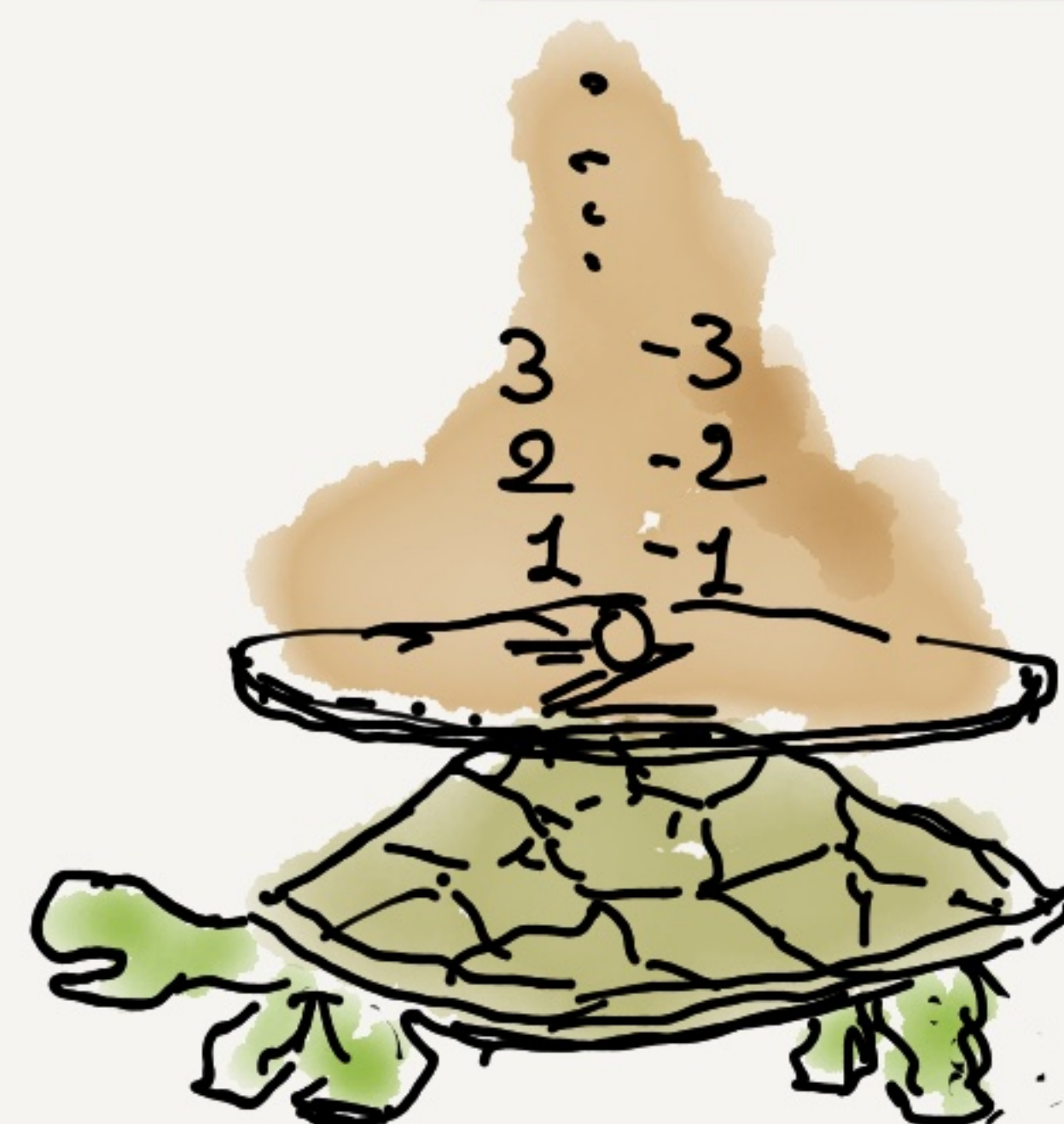
BUILDING
BLOCKS/INSTANTIATIONS



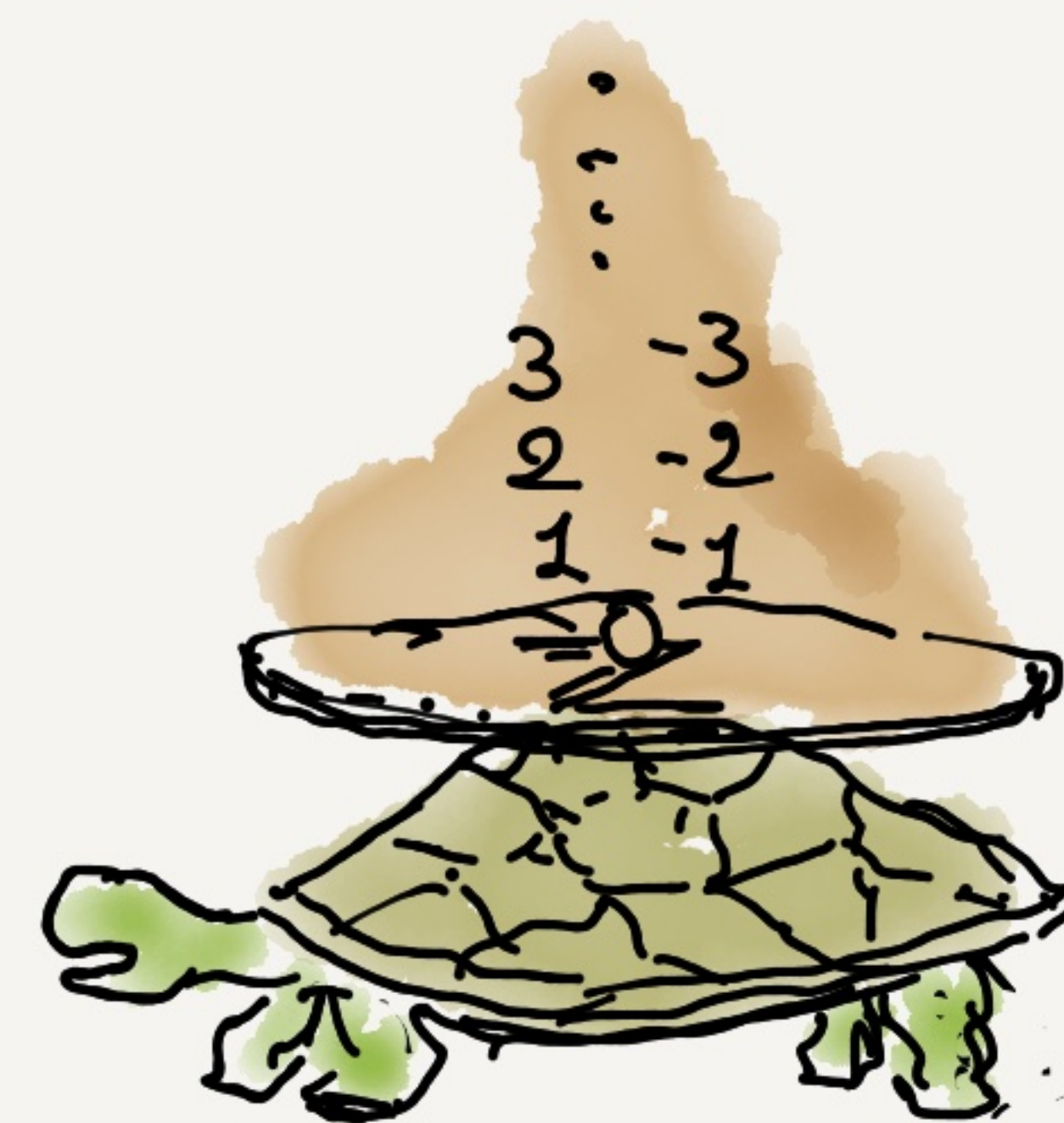
CONCRETE
CONSTRUCTION

: ZARATAN
(SPARTAN FOR Z)

FIRST
FULLY SUCCINCT SNARK FOR Z



EFFICIENCY OF ZARATAN



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

m : # OF R1CS
 \mathbb{Z} CONSTRAINTS

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

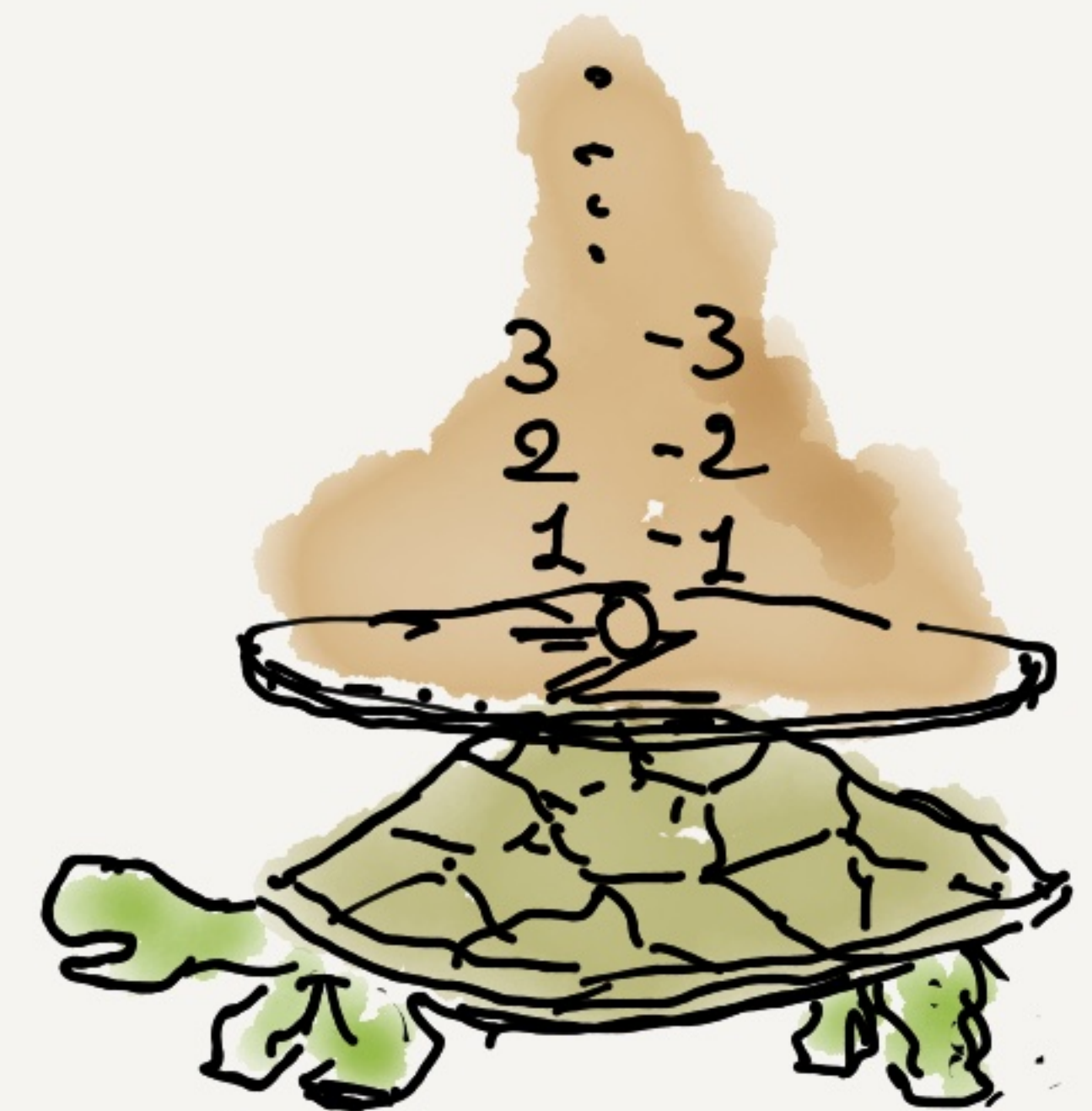
NB: $|COMPUTATION| = m \cdot m$

EFFICIENCY OF ZARATAN

TIME(V)
IS
SIMILAR

FULLY
SUCCINCT
ZARATAN

$ \Pi $	ANY COMPUTATION?	TIME(P)
$\log(m)$ $+ \log^2(m)$	SOME: $\frac{m^2}{n} < 1$	LINEAR
	ANY	"LINEAR--"



Update from Feb 15th 2025: This slide is out of date. Specifically, in a new version of the protocol the prover time never has to perform this $O(m^2)$ step anymore; its running time is always quasilinear.

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

$$\bar{w} \in \mathbb{Z}^m$$

m : # OF R1CS
CONSTRAINTS

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

NB: $|COMPUTATION| = m \cdot m$

EFFICIENCY OF ZARATAN

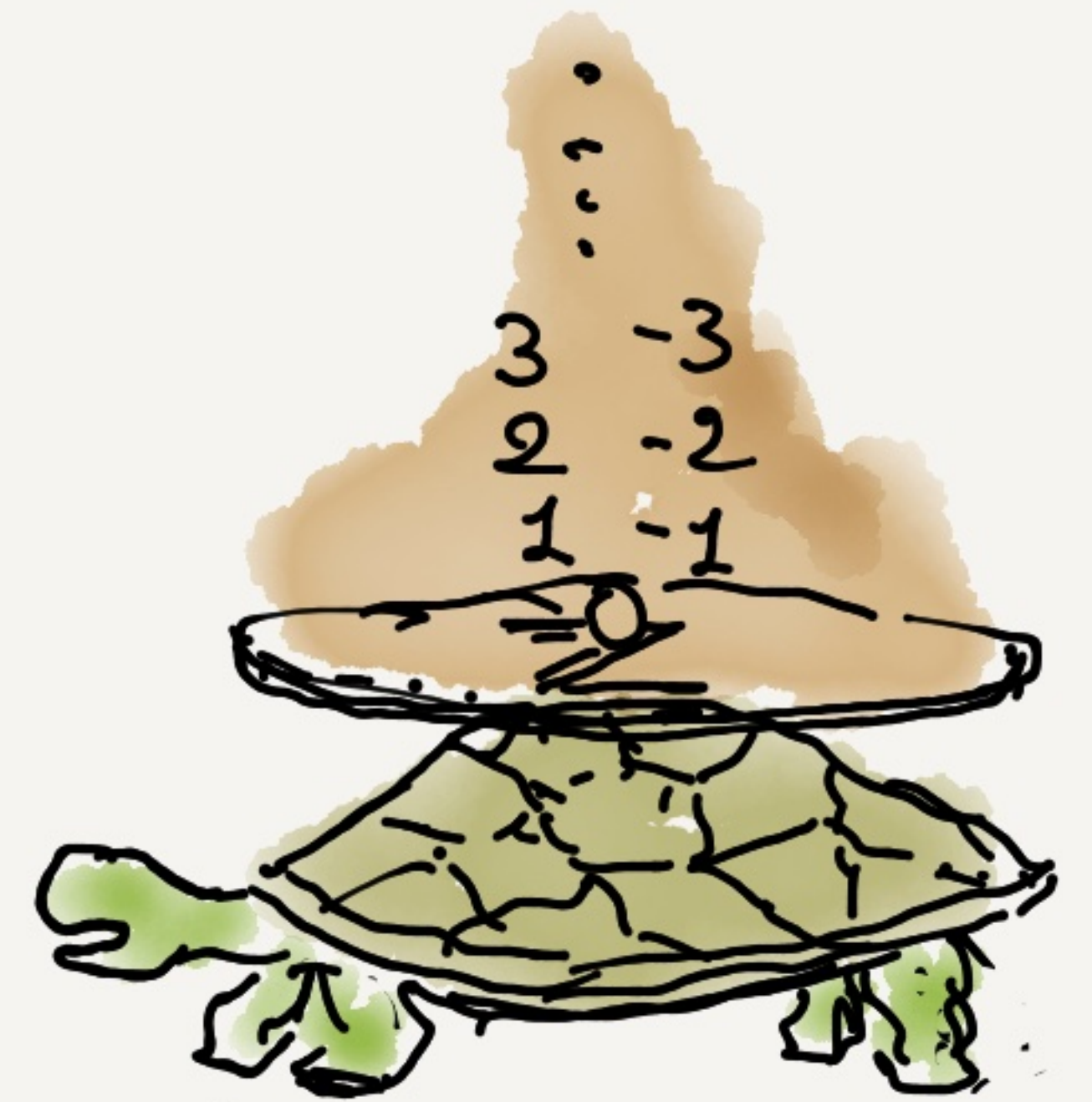
FULLY
SUCCINCT
ZARATAN

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

Update from Feb 15th 2025: This slide is out of date.
Specifically, in a new version of the protocol the
prover time never has to perform this $O(m^2)$ step
anymore; its running time is always quasilinear.

SUCCINCT
ZARATAN

(ABOVE) $+ \approx m$	ANY	LINEAR
--------------------------	-----	--------



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

$$\bar{w} \in \mathbb{Z}^m$$

m : # OF R1CS
CONSTRAINTS

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

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

② REVIEW OF TECHNIQUES

Our Core Technique:

FINGERPRINTING

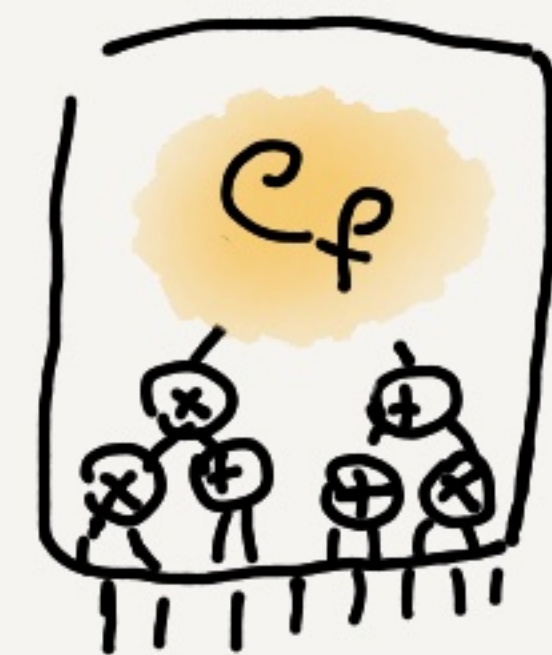


FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

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

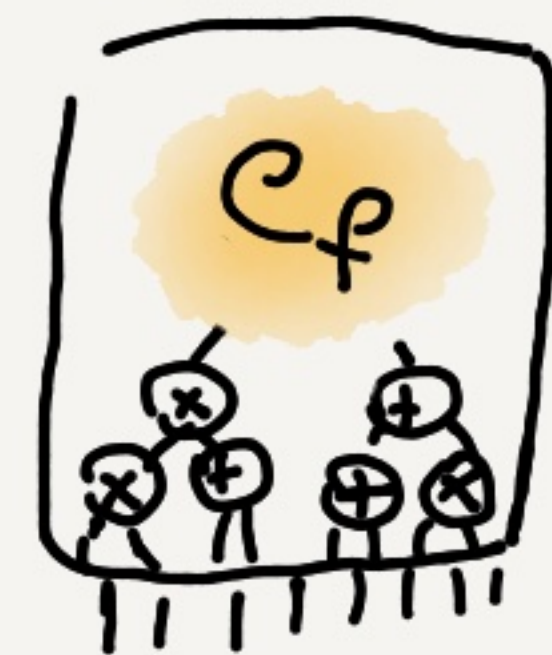


$$|C_f| = \text{poly}(n)$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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



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

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

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

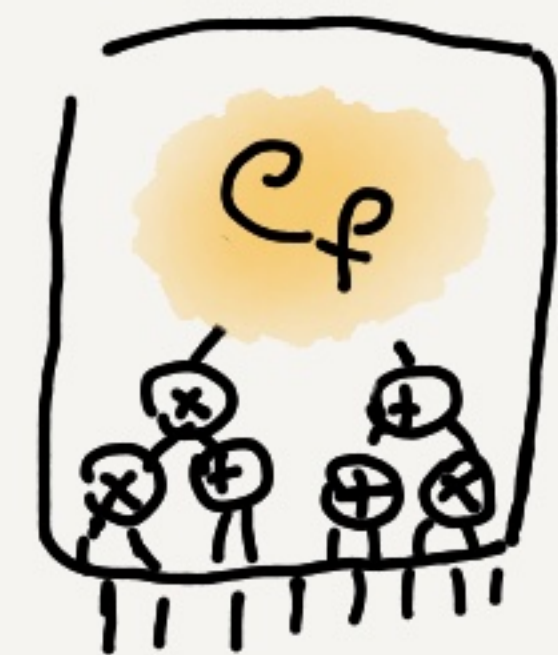
$$\Pr_{\substack{\pi_1, \dots, \pi_n \\ \leftarrow f S}} [f(\pi_1, \dots, \pi_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| return $f(r_1, \dots, r_n) \stackrel{?}{=} 0$



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

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

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

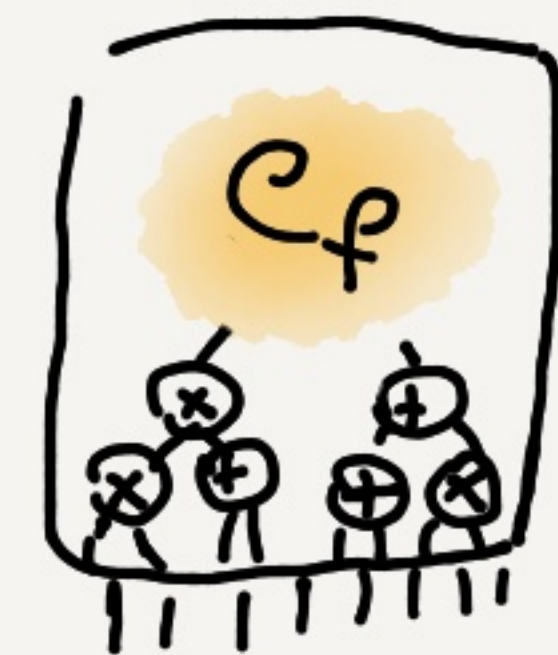
$$\Pr_{\substack{r_1, \dots, r_n \\ \leftarrow S}} [f(r_1, \dots, r_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| return $f(r_1, \dots, r_n) \stackrel{?}{=} 0$



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

\Downarrow

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

(WHY?)

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

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

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

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

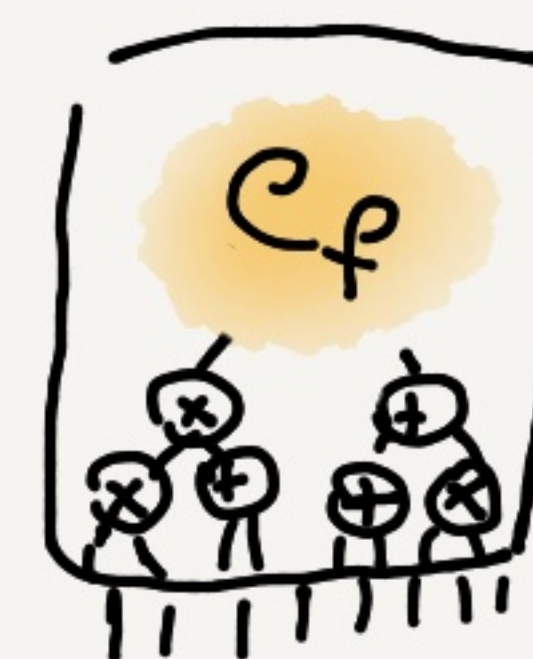
FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $\pi_1, \dots, \pi_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| RETURN $f(\pi_1, \dots, \pi_n) \stackrel{?}{=} 0$

WHAT M ? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS
 $\hookrightarrow \log(M) = \text{poly}$



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

\Downarrow

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

(WHY?)

• EACH $\otimes \Rightarrow$ DEGREE DOUBLES

• POLY DEPTH
 $\Rightarrow 2^{\text{poly DEGREE}}$

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1, \dots, x_n], f \neq 0$:
 $\Pr_{\substack{\pi_1, \dots, \pi_n \\ \leftarrow S}} [f(\pi_1, \dots, \pi_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$

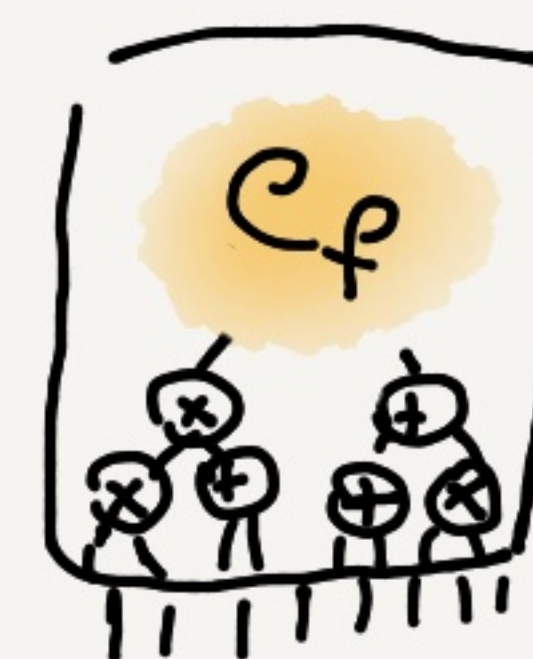
FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $\pi_1, \dots, \pi_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| RETURN $f(\pi_1, \dots, \pi_n) \stackrel{?}{=} 0$

WHAT M ? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS
 $\hookrightarrow \log(M) = \text{poly}$



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

\Downarrow

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

(WHY?)

• EACH $\otimes \Rightarrow$ DEGREE DOUBLES

• POLY DEPTH
 $\Rightarrow 2^{\text{poly DEGREE}}$

BUT... NOT POLYTIME!! WHY?

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1, \dots, x_n], f \neq 0$:
 $\Pr_{\substack{\pi_1, \dots, \pi_n \\ \leftarrow S}} [f(\pi_1, \dots, \pi_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$

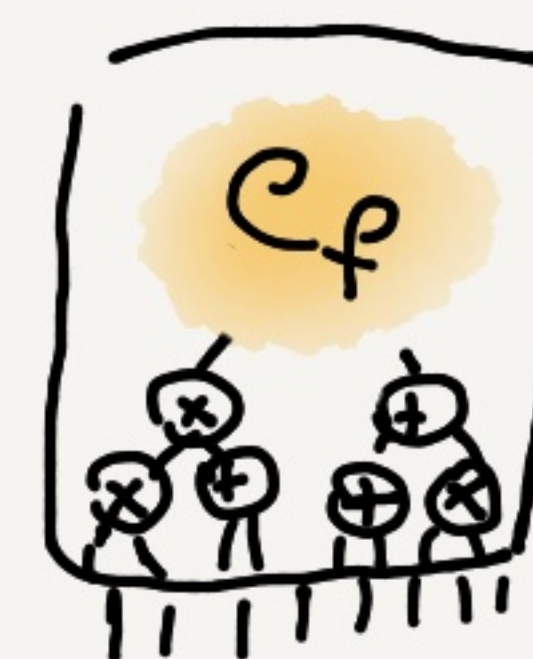
FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $\pi_1, \dots, \pi_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| RETURN $f(\pi_1, \dots, \pi_n) \stackrel{?}{=} 0$

WHAT M ? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS
 $\hookrightarrow \log(M) = \text{poly}$



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

\Downarrow

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

(WHY?)

• EACH $\otimes \Rightarrow$ DEGREE DOUBLES

• POLY DEPTH
 $\Rightarrow 2^{\text{poly DEGREE}}$

BUT... NOT POLYTIME!! WHY?

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

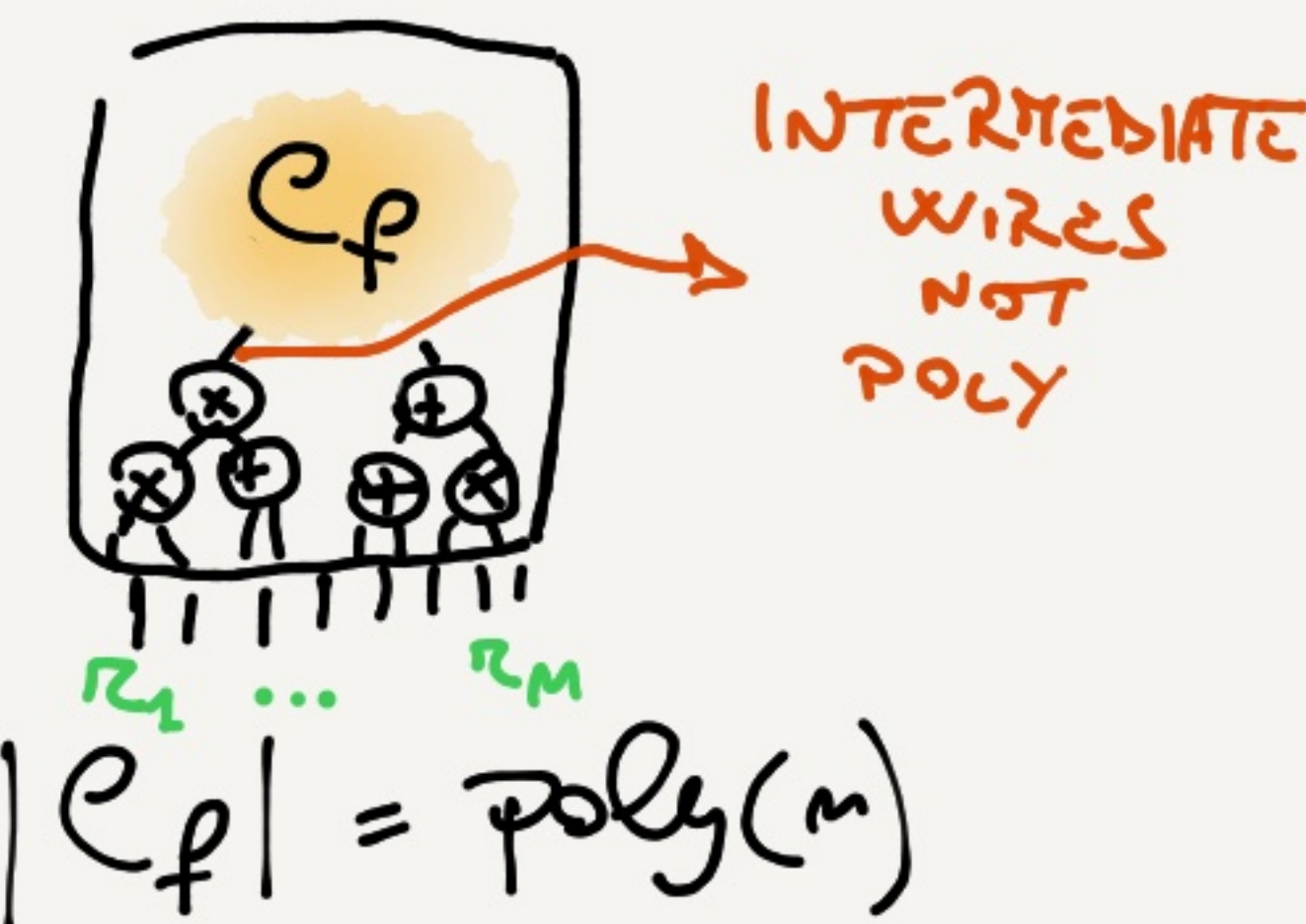
LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1, \dots, x_n], f \neq 0$:
 $\Pr_{\substack{\pi_1, \dots, \pi_n \\ \leftarrow S}} [f(\pi_1, \dots, \pi_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $\pi_1, \dots, \pi_n \leftarrow \{1, \dots, M\}$ (FOR AN M WE WILL PICK)
| RETURN $f(\pi_1, \dots, \pi_n) \stackrel{?}{=} 0$
WHAT M ? FOR $O(1)$ ERROR, $M \approx 100 \cdot \deg(f)$ WORKS
 $\hookrightarrow \log(M) = \text{poly}$ ✓



BUT... NOT POLY TIME!! WHY?

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

(WHY?)

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

IMPLIES
BIT SIZE
DOUBLES

TOOL: GENERALIZED SCHWARTZ-ZIPPEL LEMMA

LET $S \subseteq \mathbb{Z}$. $\forall f \in \mathbb{Z}[x_1, \dots, x_n], f \neq 0$:
 $\Pr_{\substack{\pi_1, \dots, \pi_n \\ \leftarrow S}} [f(\pi_1, \dots, \pi_n) \neq 0] \geq 1 - \frac{\deg(f)}{|S|}$

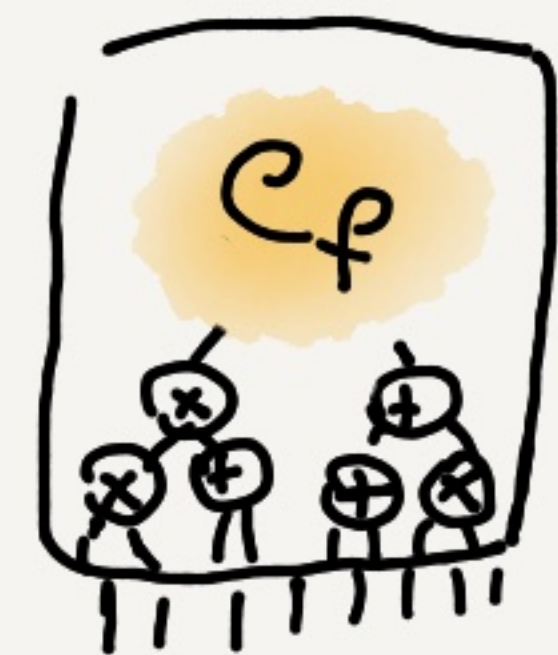
FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

APPROACH: | SAMPLE $\pi_1, \dots, \pi_n \leftarrow \{1, \dots, \mathcal{M}\}$ // $\mathcal{M} \approx 100 \cdot \deg(f)$
| return $f(\pi_1, \dots, \pi_n) \stackrel{?}{=} 0$

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



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

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

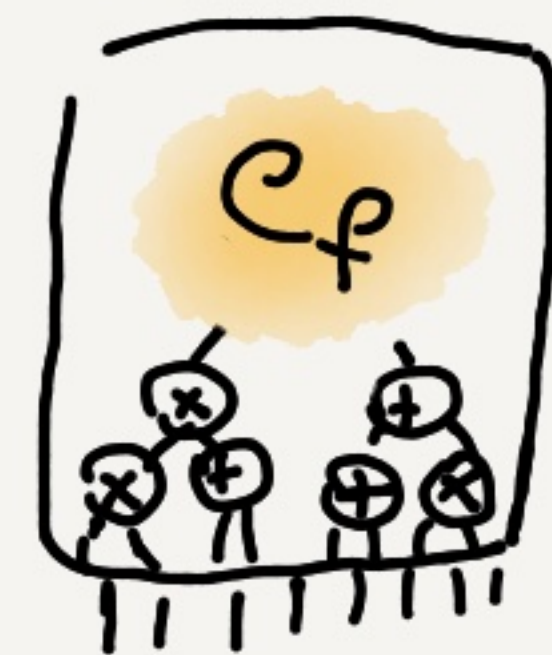
FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f

OUTPUT: IS f THE ZERO POLY?

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

APPROACH: | SAMPLE $r_1, \dots, r_n \leftarrow \{1, \dots, M\}$ // $M \approx 100 \cdot \deg(f)$
| return $\frac{f(r_1, \dots, r_n) \stackrel{?}{=} 0 \pmod{q}}{(FOR\ RANDOM\ PRIME\ OF\ L\ BITS)}$



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

$$\Downarrow$$

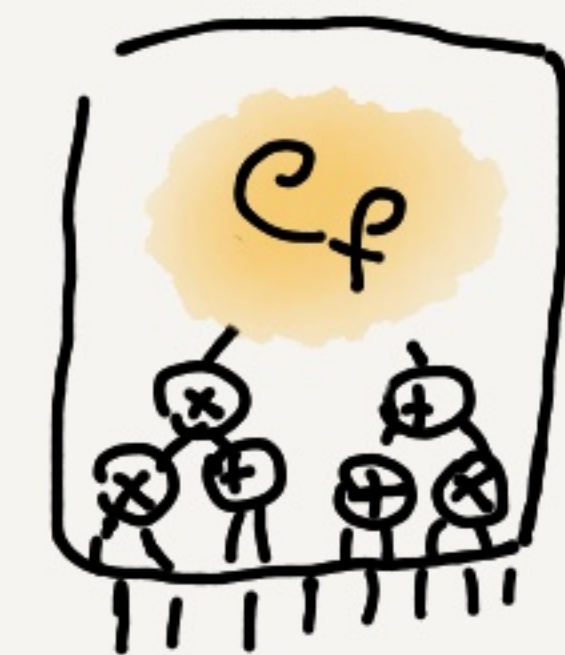
$$\deg(f) = O(2^{n^c})$$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

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)



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



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

WHY DOES THIS WORK?

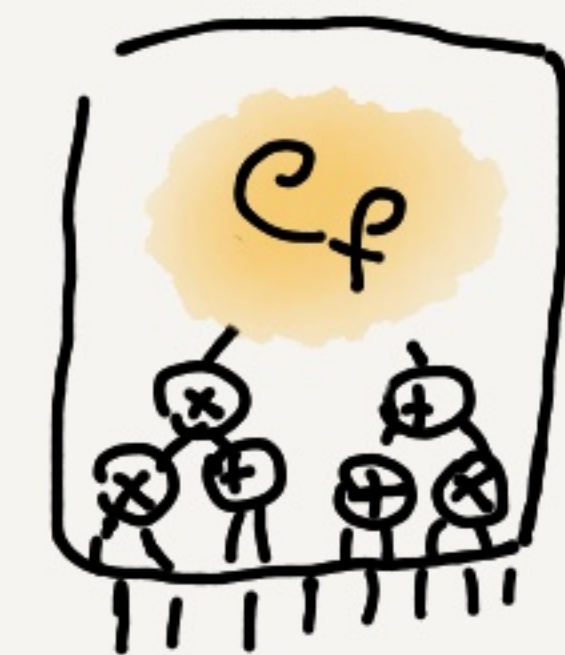
• IF $f(r_1, \dots, r_n) = 0$ THEN $f(r_1, \dots, r_n) \equiv 0 \pmod{q}$ ✓

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)



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



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

WHY DOES THIS WORK?

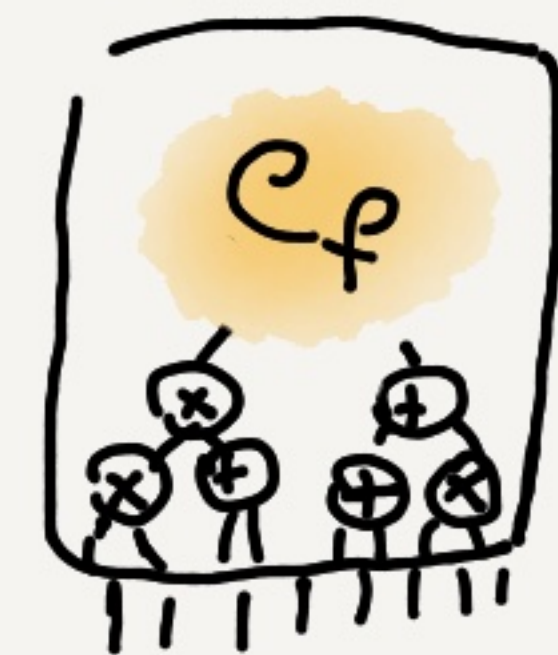
- IF $f(r_1, \dots, r_n) = 0$ THEN $f(r_1, \dots, r_n) \equiv 0 \pmod{q}$ ✓
- IF $\underbrace{f(r_1, \dots, r_n)}_y \neq 0$ THEN $f(r_1, \dots, r_n) \not\equiv 0 \pmod{q}$
UNLESS $q|y$

FINGERPRINTING THROUGH AN EXAMPLE: ZERO-TESTING POLYNOMIALS

INPUT: CIRCUIT C_f COMPUTING f
OUTPUT: IS f THE ZERO POLY?

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

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)



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



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

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 $q|y$

equivalent to:

$$q \in \{p_1 \dots p_t\} \text{ w/ } y = p_1^{x_1} \dots p_t^{x_t}$$

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

FINGERPRINTING FOR OUR SETTING*

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

AN ATTEMPT TO A TEMPLATE

$C(\bar{x})=0$
↓
PROVER ($\bar{x} \in \mathbb{Z}^m$)

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

VERIFIER (vk_C)

AN ATTEMPT TO A TEMPLATE

$C(\bar{x})=0$
PROVER ($\bar{x} \in \mathbb{Z}^m$)

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

VERIFIER (vk_C)

INTUITION(S) WE WILL
TRY* AND LEVERAGE:



LET US
"FINGERPRINT"
COMPUTATION
 C NOW

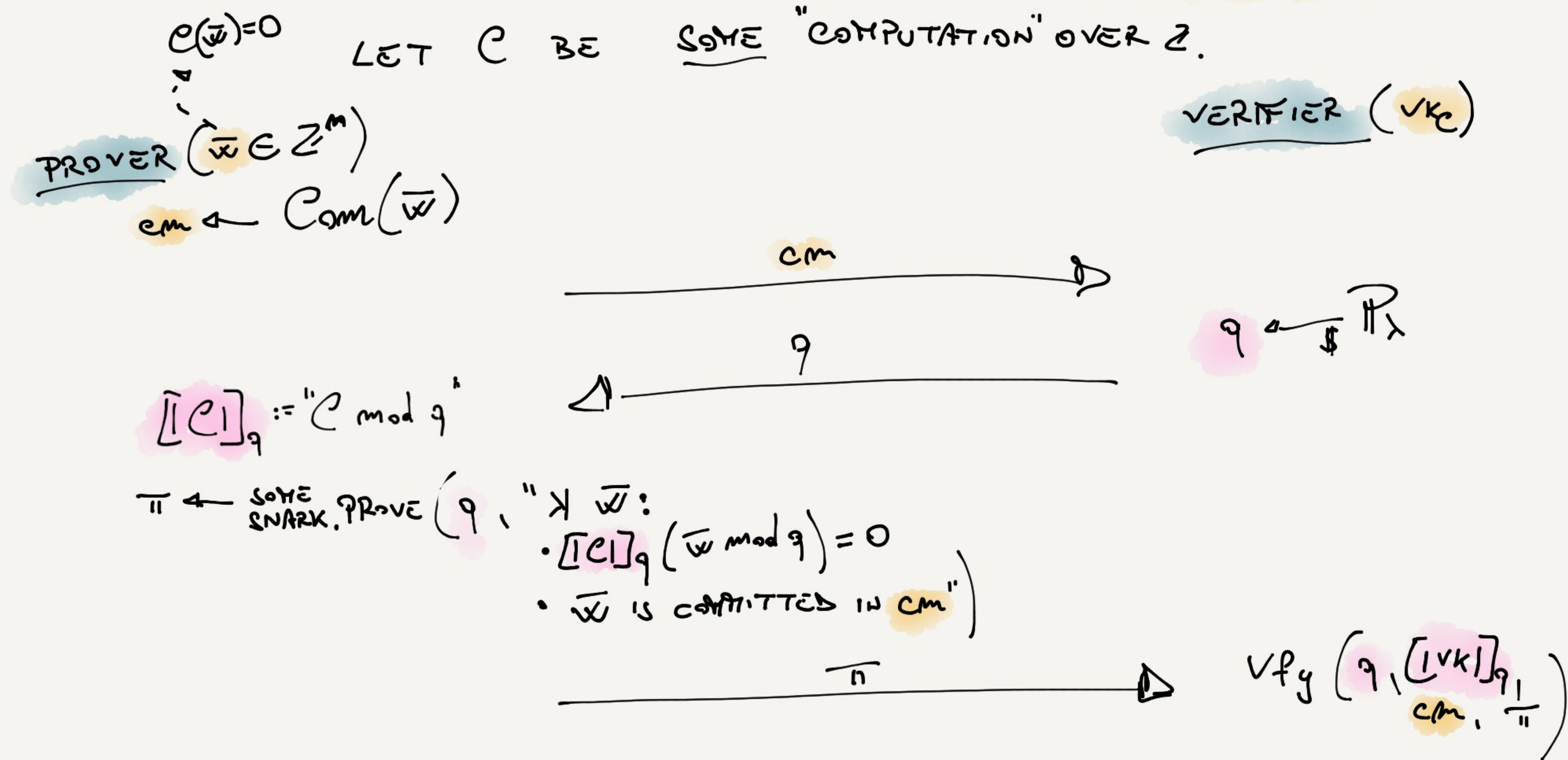
MAYBE WE CAN
USE A SNARK
"MOD 9"...

PRIME TOR

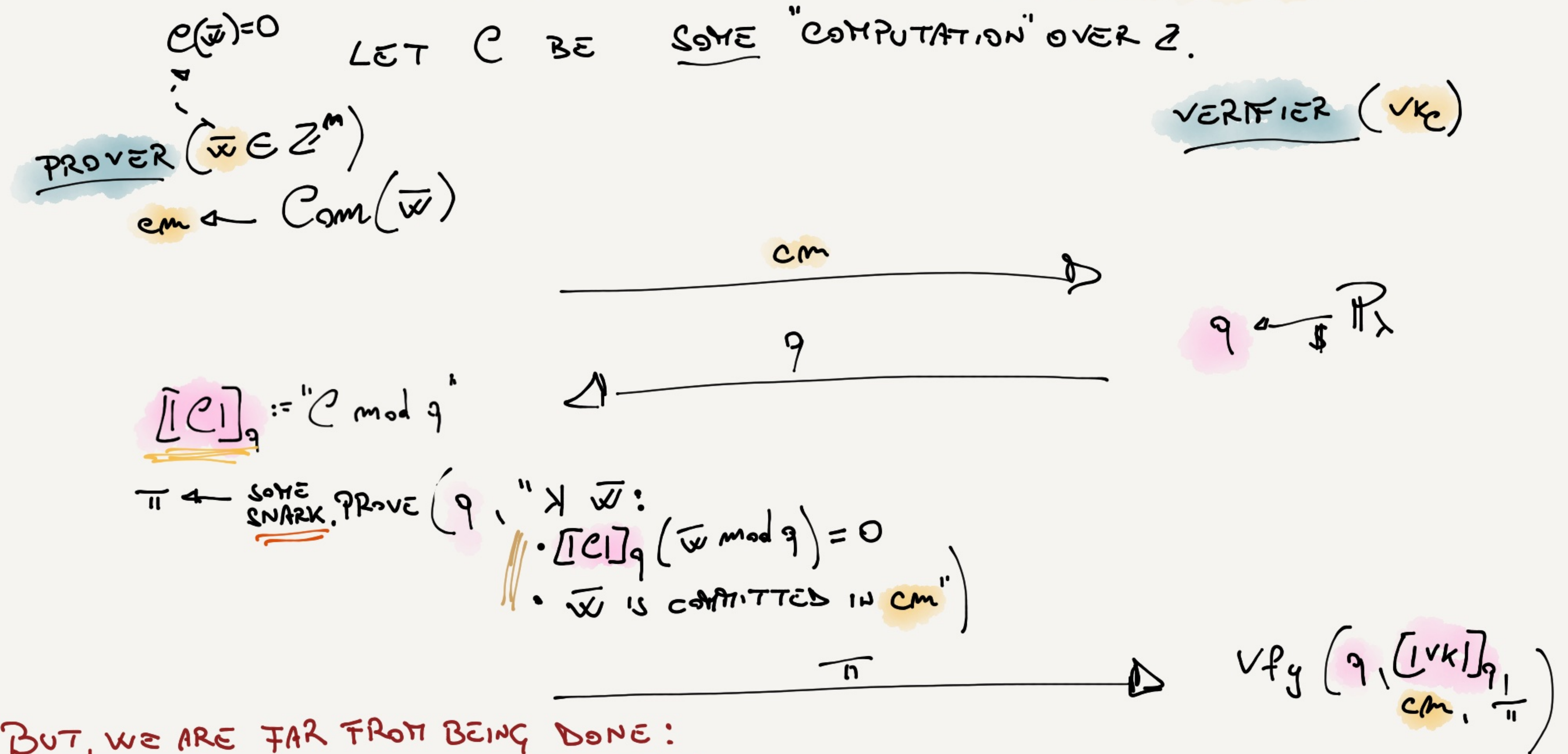


* AND IN GENERAL THIS WILL FAIL
(PLUS WILL STILL REQUIRE MORE WORK)

AN ATTEMPT TO A TEMPLATE



AN ATTEMPT TO A TEMPLATE

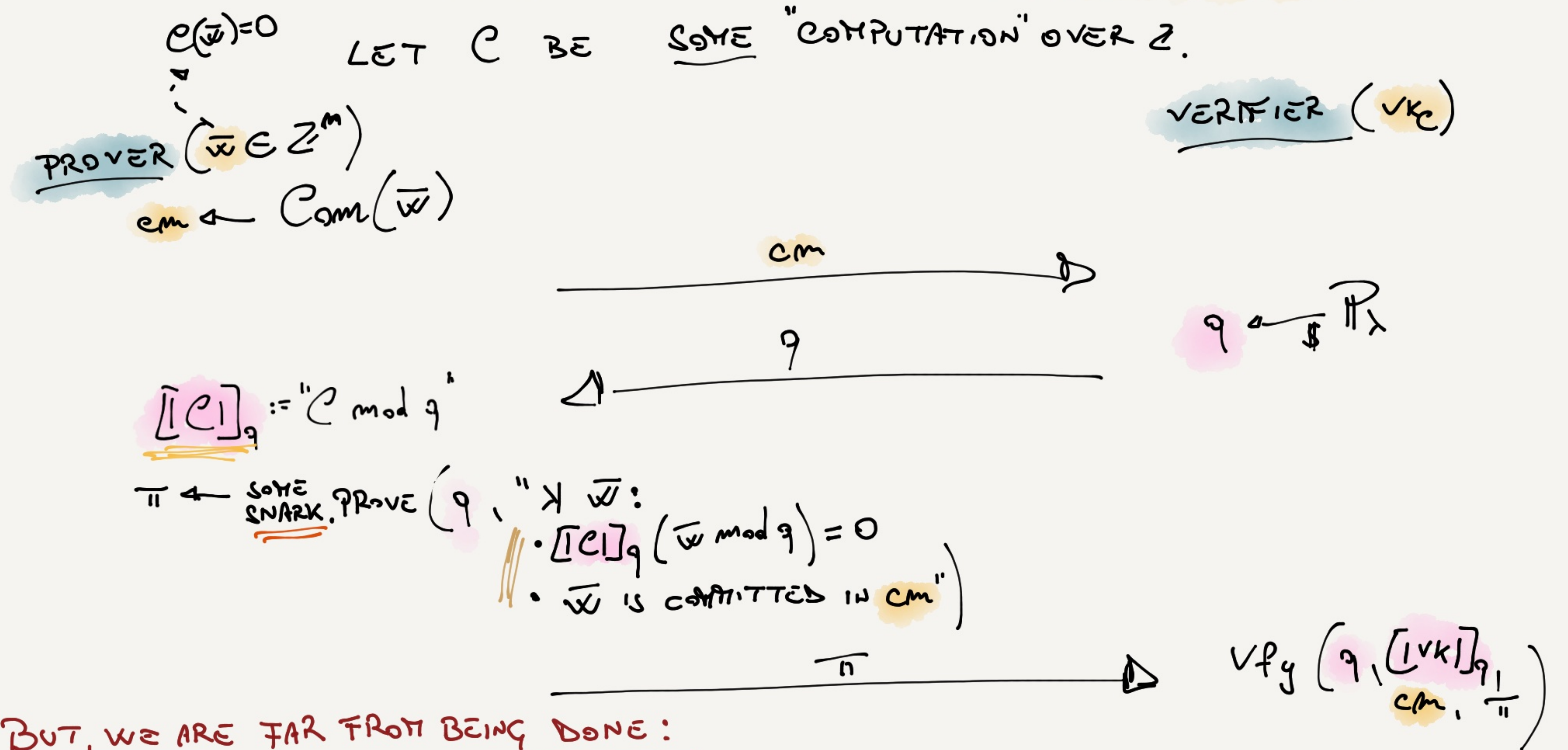


BUT, WE ARE FAR FROM BEING DONE:

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

● : BEFORE SAMPLING q
● : AFTER SAMPLING q

AN ATTEMPT TO A TEMPLATE



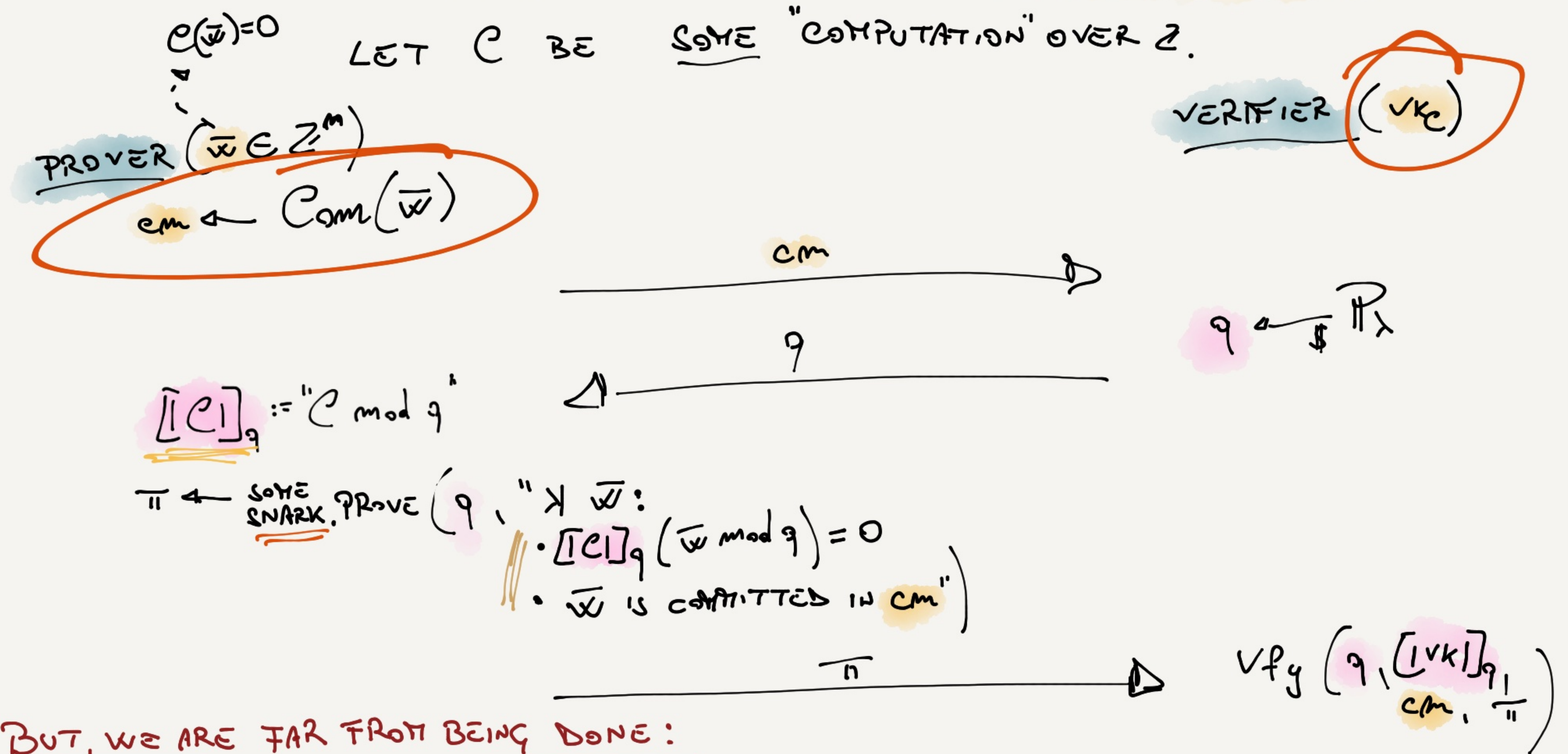
BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOME SNARK"? WHAT PROPERTIES NECESSARY?

CHALLENGES:

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

AN ATTEMPT TO A TEMPLATE



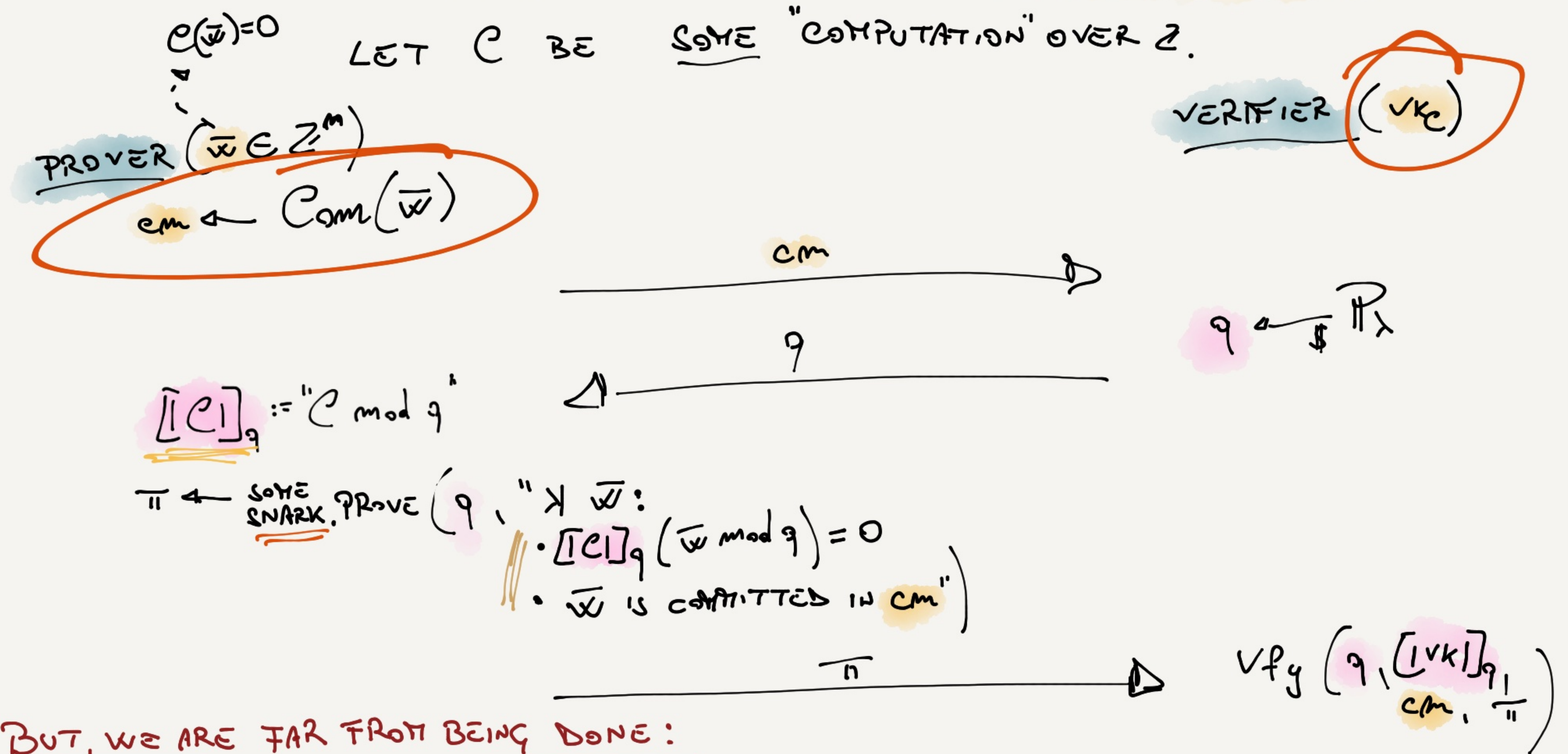
BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOME SNARK"? WHAT PROPERTIES NECESSARY?

CHALLENGES:

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

AN ATTEMPT TO A TEMPLATE



BUT, WE ARE FAR FROM BEING DONE:

1) WHAT IS "SOME SNARK"? WHAT PROPERTIES NECESSARY?

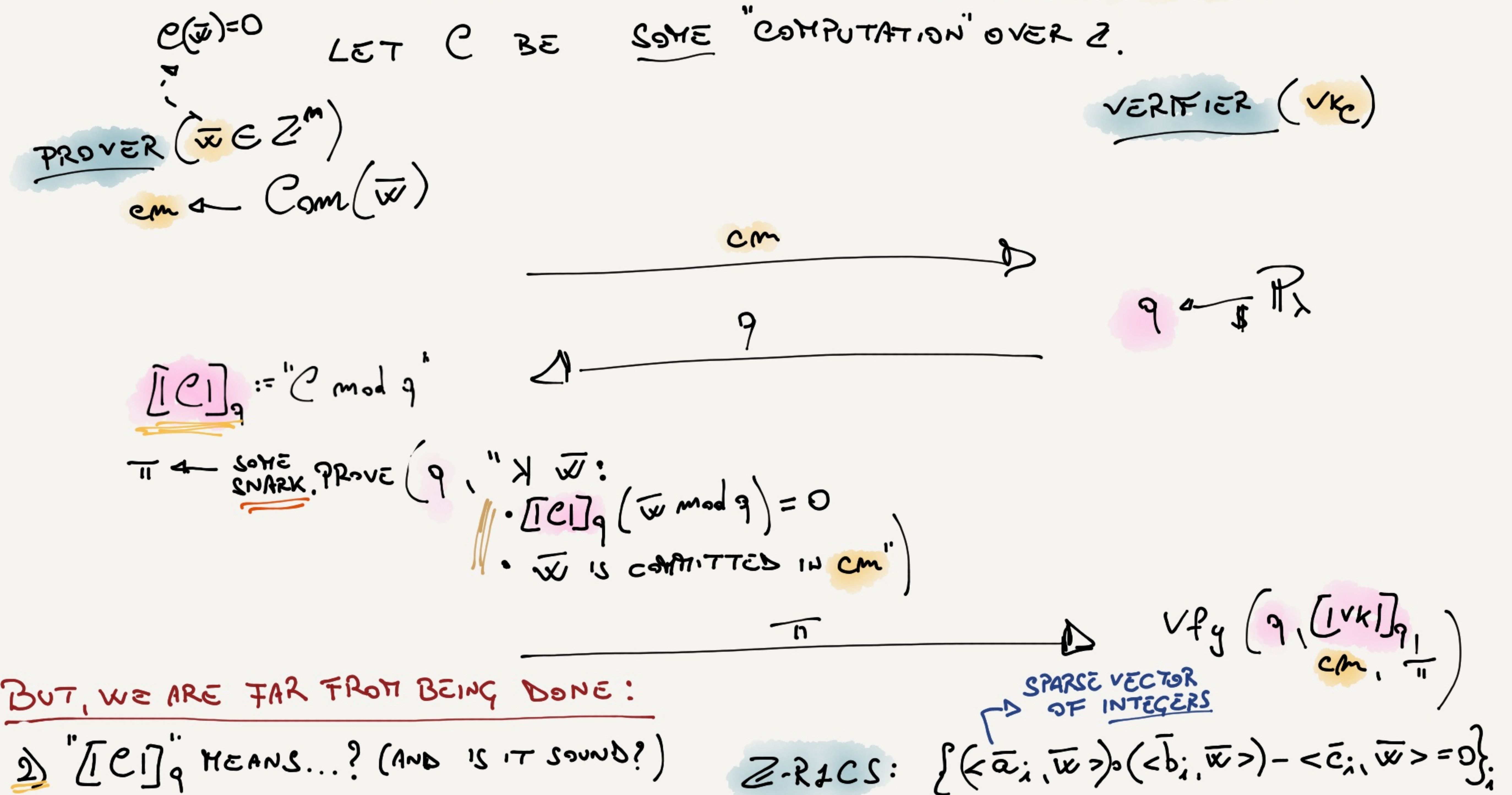
CHALLENGES:

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

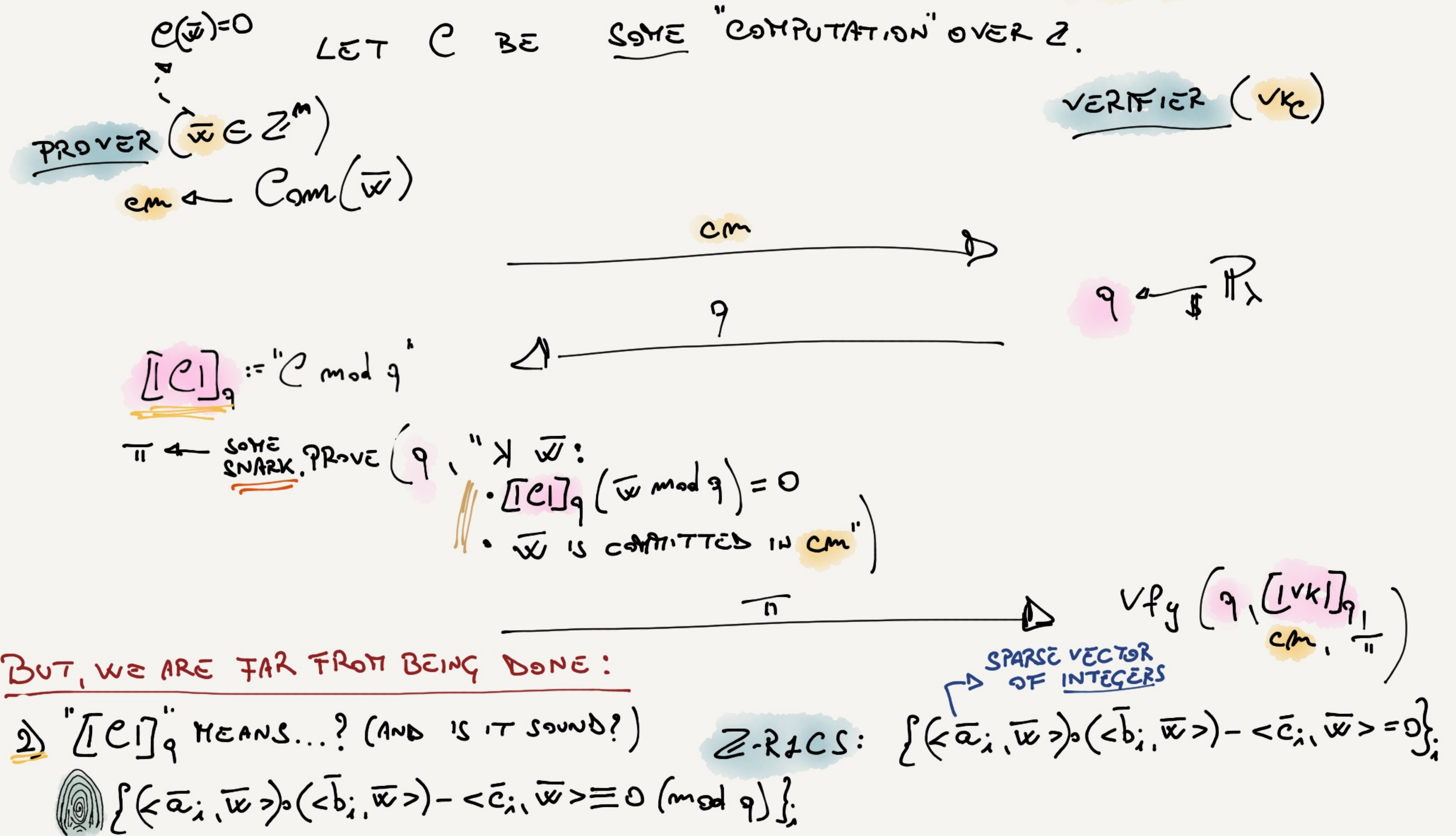
SCHEMES BASED ON THE GOOD CANDIDATES (SPARTAN, HYPERPLONK, ...)

↳ MULTI-LINEAR
EXTENSIONS

AN ATTEMPT TO A TEMPLATE



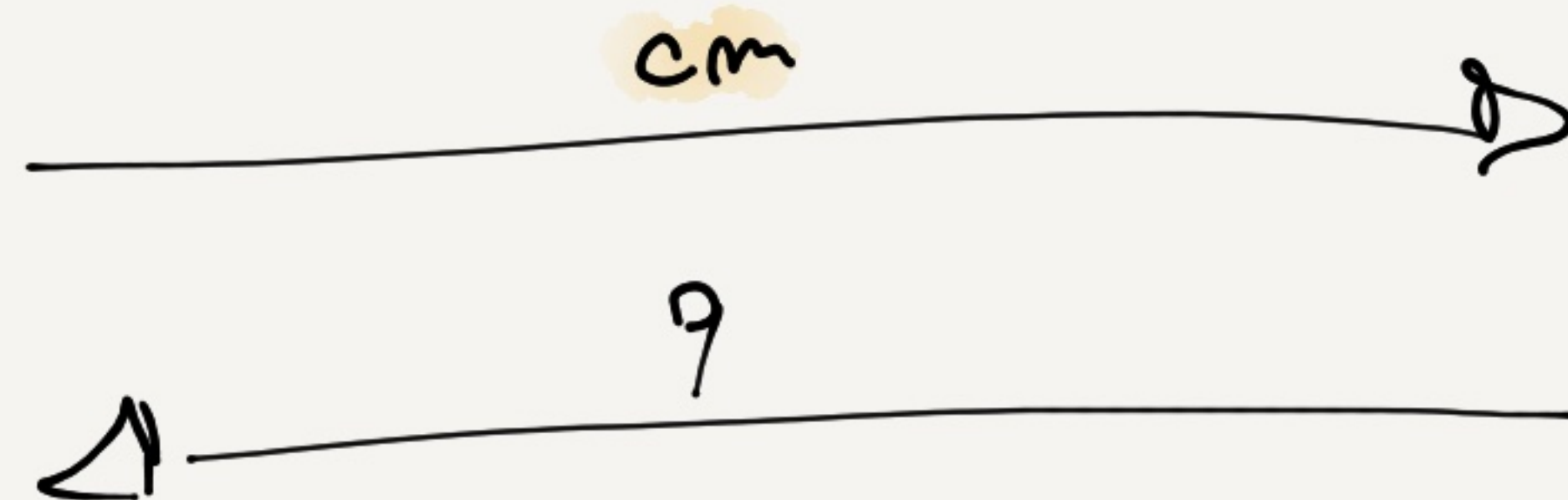
AN ATTEMPT TO A TEMPLATE



AN ATTEMPT TO A TEMPLATE

$C(\bar{w}) = 0$
 LET C BE SOME "COMPUTATION" OVER \mathbb{Z} .
 PROVER ($\bar{w} \in \mathbb{Z}^m$)
 $cm \leftarrow Com(\bar{w})$

VERIFIER (vk_c)



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

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

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

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

BUT, WE ARE FAR FROM BEING DONE:

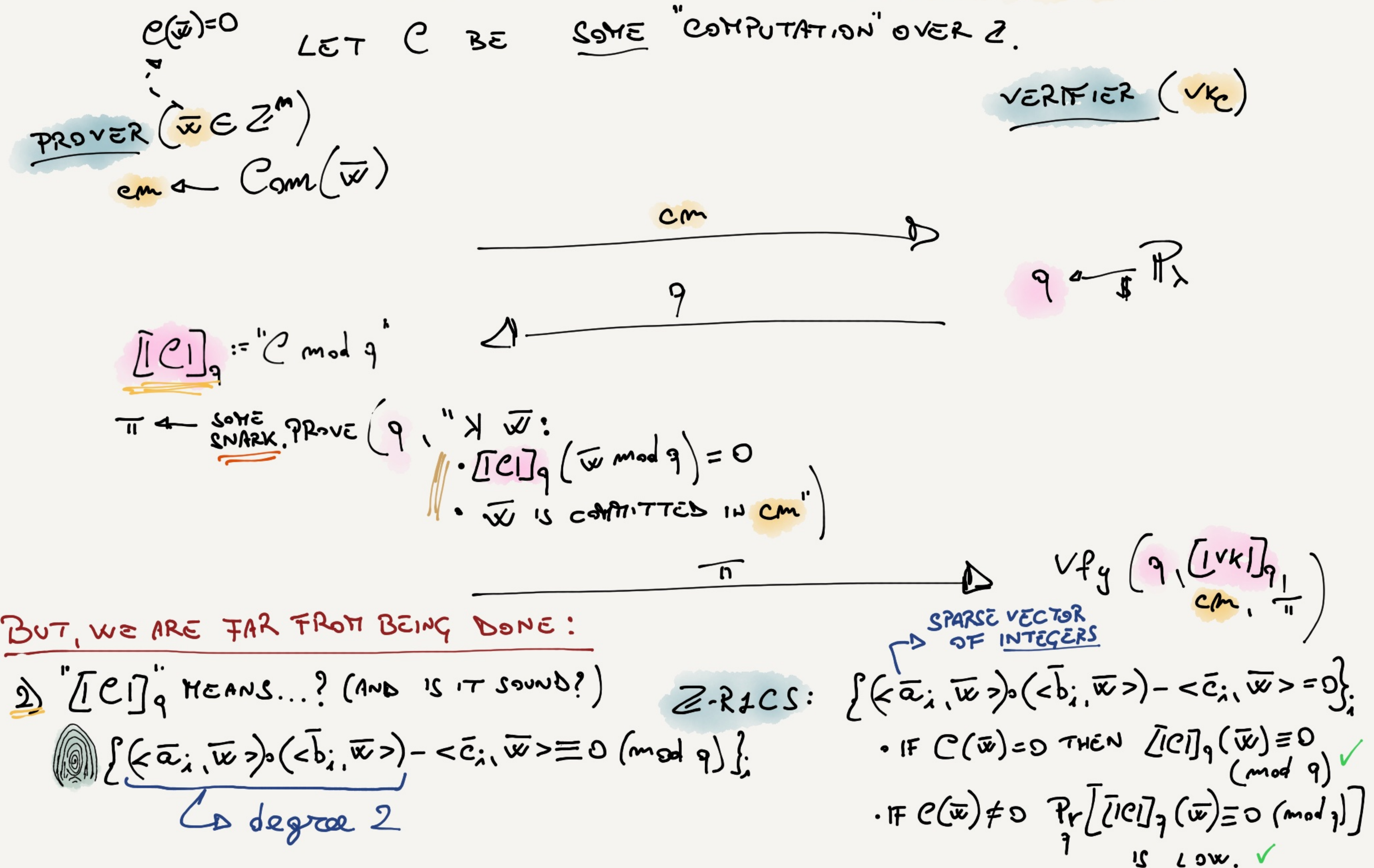
1) $[C]_q$ MEANS...? (AND IS IT SOUND?)

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

$\mathbb{Z}\text{-R1CS: } \{ \langle \bar{a}_i, \bar{w} \rangle \cdot \langle \bar{b}_i, \bar{w} \rangle - \langle \bar{c}_i, \bar{w} \rangle = 0 \}_i$
 \bullet IF $C(\bar{w}) = 0$ THEN $[C]_q(\bar{w}) \equiv 0 \pmod{q}$ ✓

SPARSE VECTOR OF INTEGERS

AN ATTEMPT TO A TEMPLATE



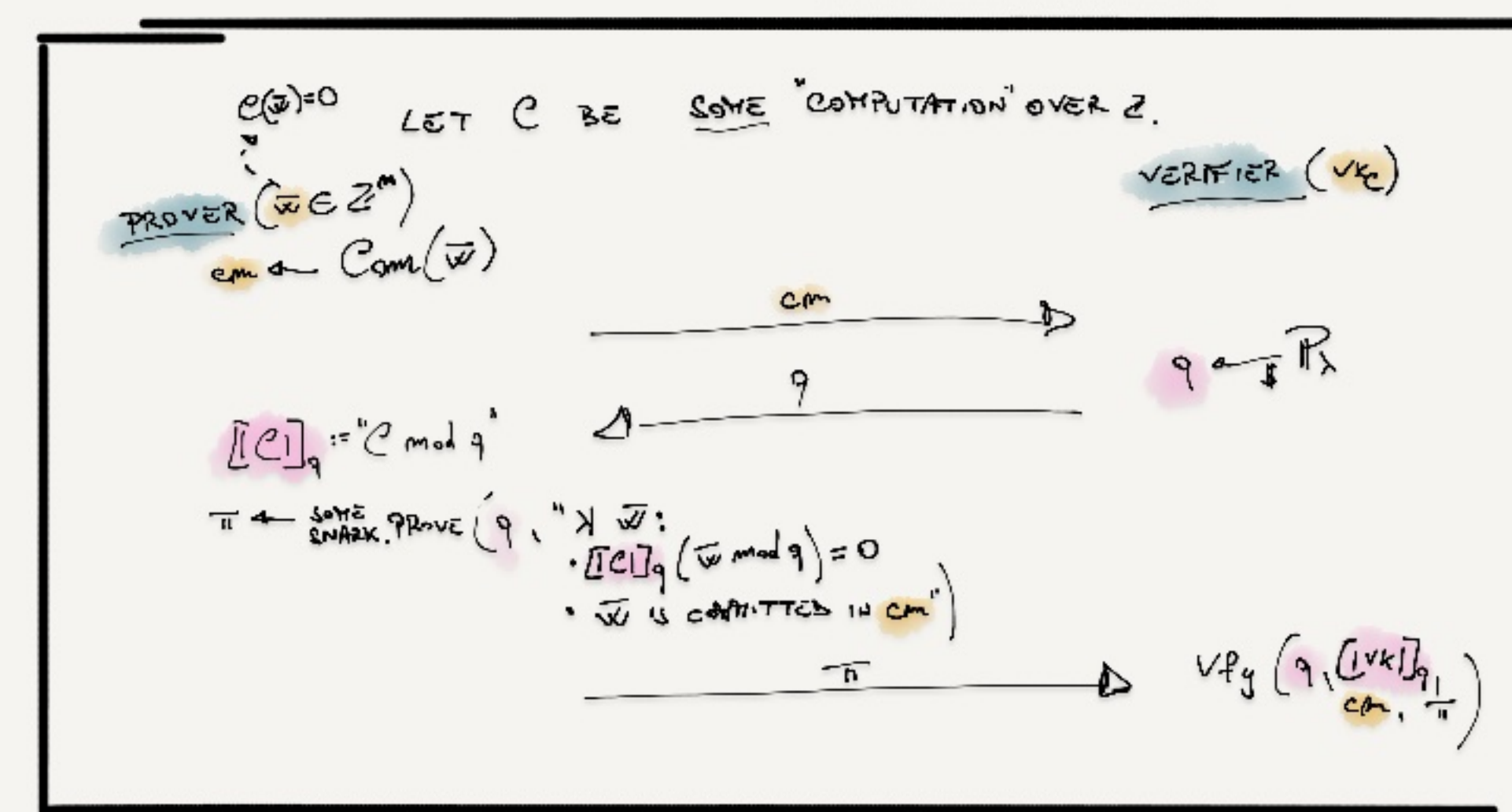
JUST NOW:

FLAVOR OF THE SOLUTION + CHALLENGES

NEXT:

THE ACTUAL FRAMEWORK

↓
(SPOILER: VARIANT OF AHP+PC)

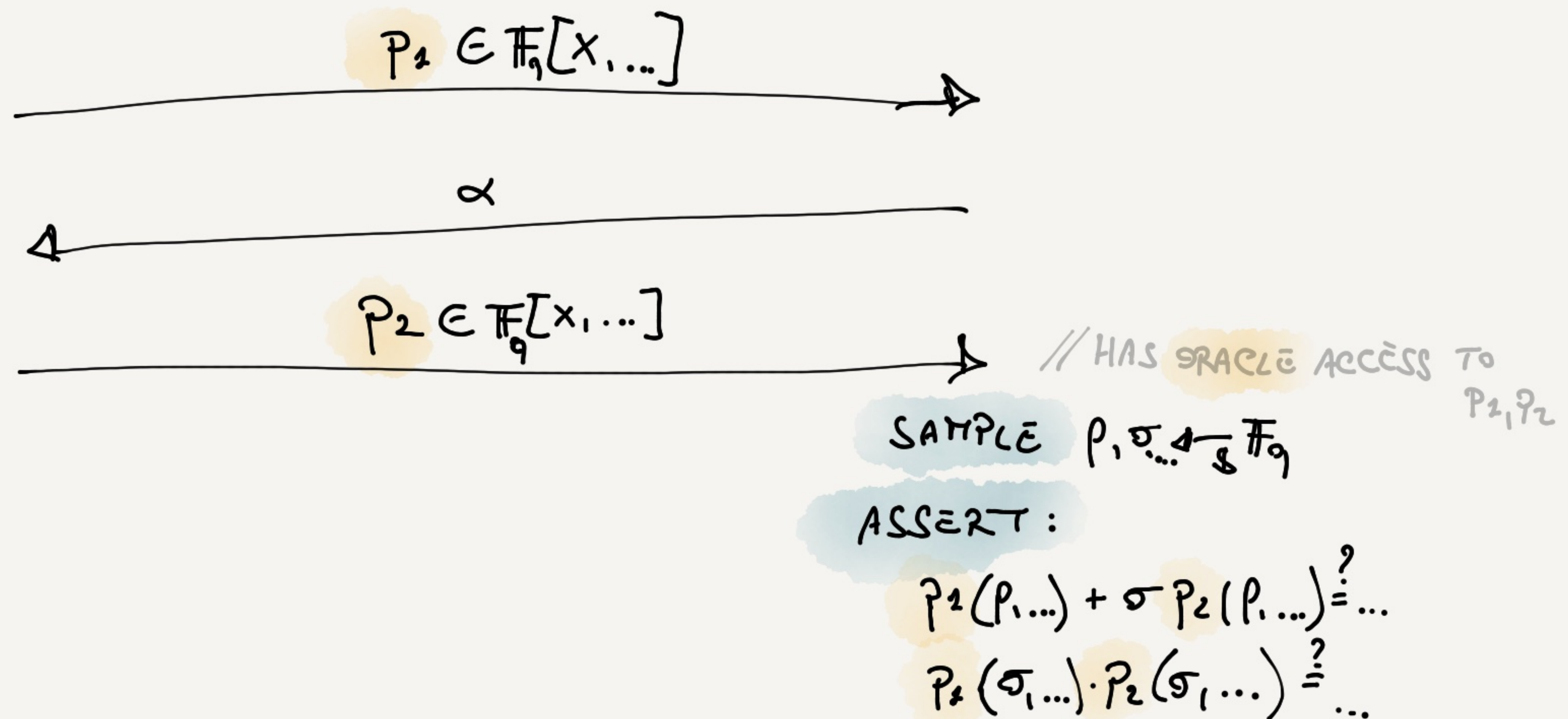


IMPLICITLY CAPTURED HERE

BRUSH UP ON AHPs / P1OP2 / PHPs* [MARLIN, DARK, PLONK, LUNAR, ...]

PROVER ^{\mathbb{F}_q} (w)

VERIFIER ^{\mathbb{F}_q}



* ALGEBRAIC HOLOGRAPHIC PROOFS
POLYNOMIAL IOPs
POLYNOMIAL HOLOGRAPHIC IOPs

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

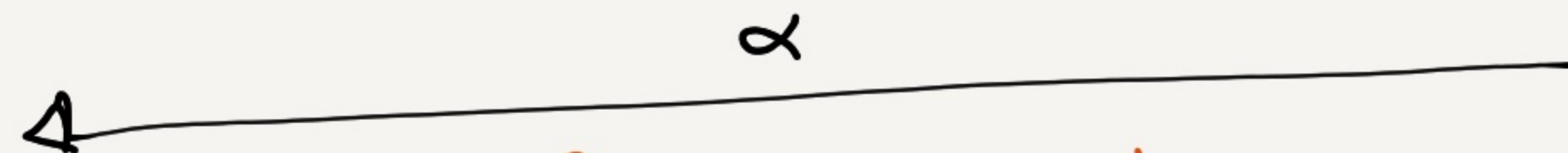
VIA POLY COMMITMENTS

PROVER \mathbb{F}_q (w)

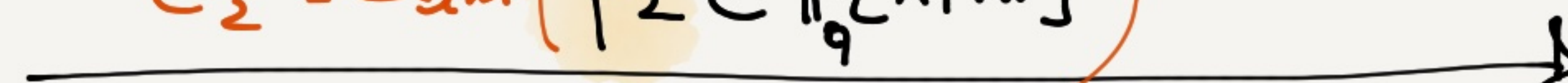
VERIFIER \mathbb{F}_q

POLYNOMIAL COMMITMENT

$$c_1 = \text{Com}(p_1 \in \mathbb{F}_q[x, \dots])$$



$$c_2 = \text{Com}(p_2 \in \mathbb{F}_q[x, \dots])$$



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

SAMPLE $p, \sigma \leftarrow \mathbb{F}_q$

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

$$p_1(p, \dots) + \sigma p_2(p, \dots) \stackrel{?}{=} \dots$$

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

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

$$a_1, a_2, b_1, b_2, \dots$$

$$\frac{(a)}{\pi_1}, \frac{(a)}{\pi_2}, \frac{(b)}{\pi_1}, \frac{(b)}{\pi_2}$$

PolyComm.Verify($c_1, (p_1, \dots)$
 $c_2, \frac{(a)}{\pi_1}$)

// ... CHECK OTHER PROOFS.

mod-AHPs: AHPs OVER \mathbb{Z} w/ MODULAR REMAINDER QUERIES

PROVER \mathbb{F}_q \swarrow VECTOR OF INTEGERS
(w)

VERIFIER \mathbb{F}_q

$z \in \mathbb{Z}[x_1, \dots]$
 q

$q \leftarrow \$ \mathbb{P}_2$

$p_1 \in \mathbb{F}_q[x_1, \dots]$

α

$p_2 \in \mathbb{F}_q[x_1, \dots]$

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

SAMPLE $\rho, \sigma \leftarrow \$ \mathbb{F}_q$

ASSERT:

$z(\rho, \dots) \cdot p_1(\rho, \dots) + \sigma p_2(\rho, \dots) \stackrel{?}{=} \dots (\text{mod } q)$
 $p_1(\sigma, \dots) \cdot p_2(\sigma, \dots) \stackrel{?}{=} \dots (\text{mod } q)$

mod-AHPs: AHPs OVER \mathbb{Z} w/ MODULAR REMAINDER QUERIES

PROVER \mathbb{F}_q \swarrow VECTOR OF INTEGERS
(w)

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

q

$$p_1 \in \mathbb{F}_q[x_1, \dots]$$

α

$$p_2 \in \mathbb{F}_q[x_1, \dots]$$

VERIFIER \mathbb{F}_q

$$q \leftarrow \$ \mathbb{P}_2$$



// HAS ORACLE ACCESS TO z, p_1, p_2
SAMPLE $p, \sigma \leftarrow \$ \mathbb{F}_q$ (mod q)

ASSERT:

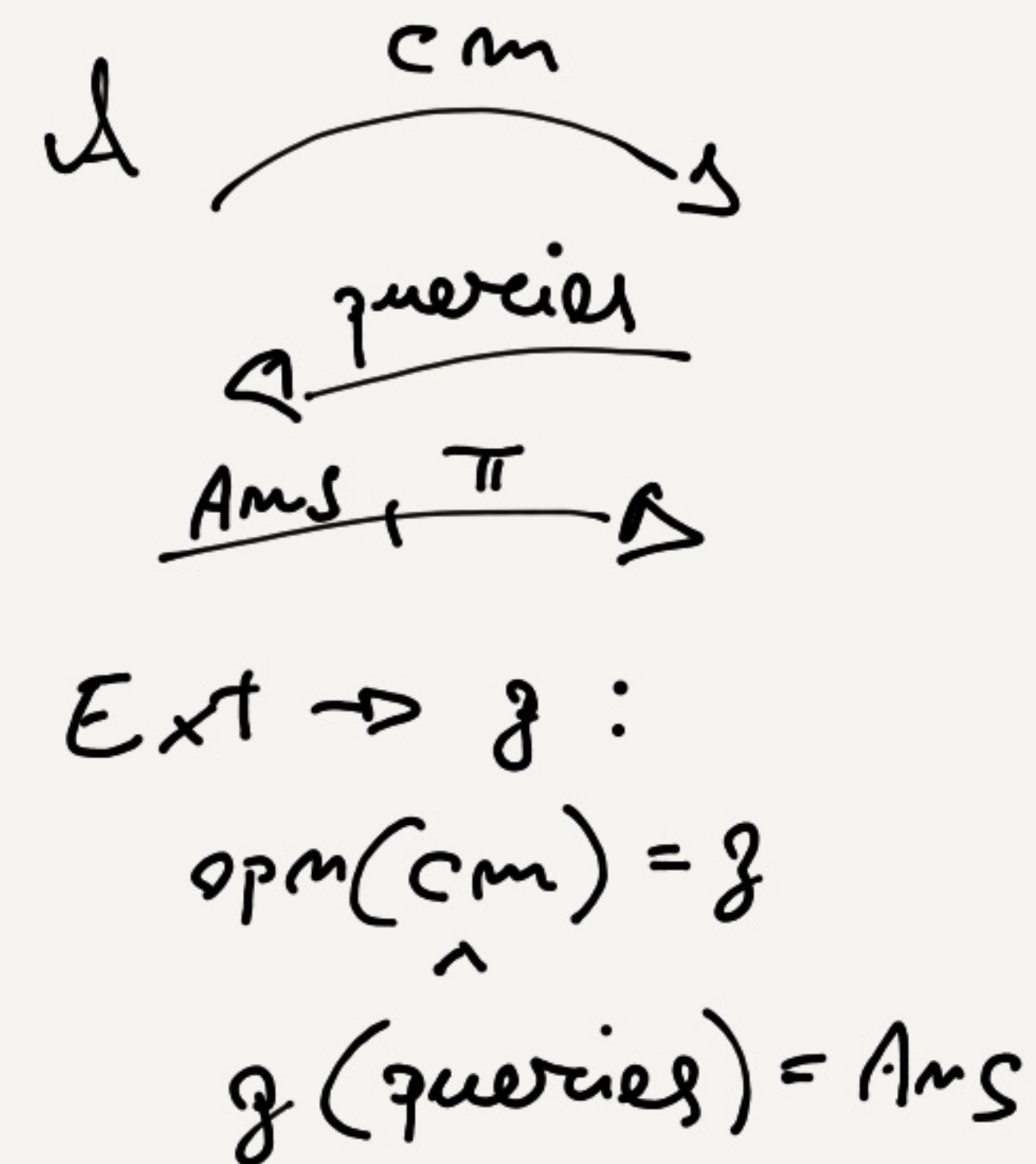
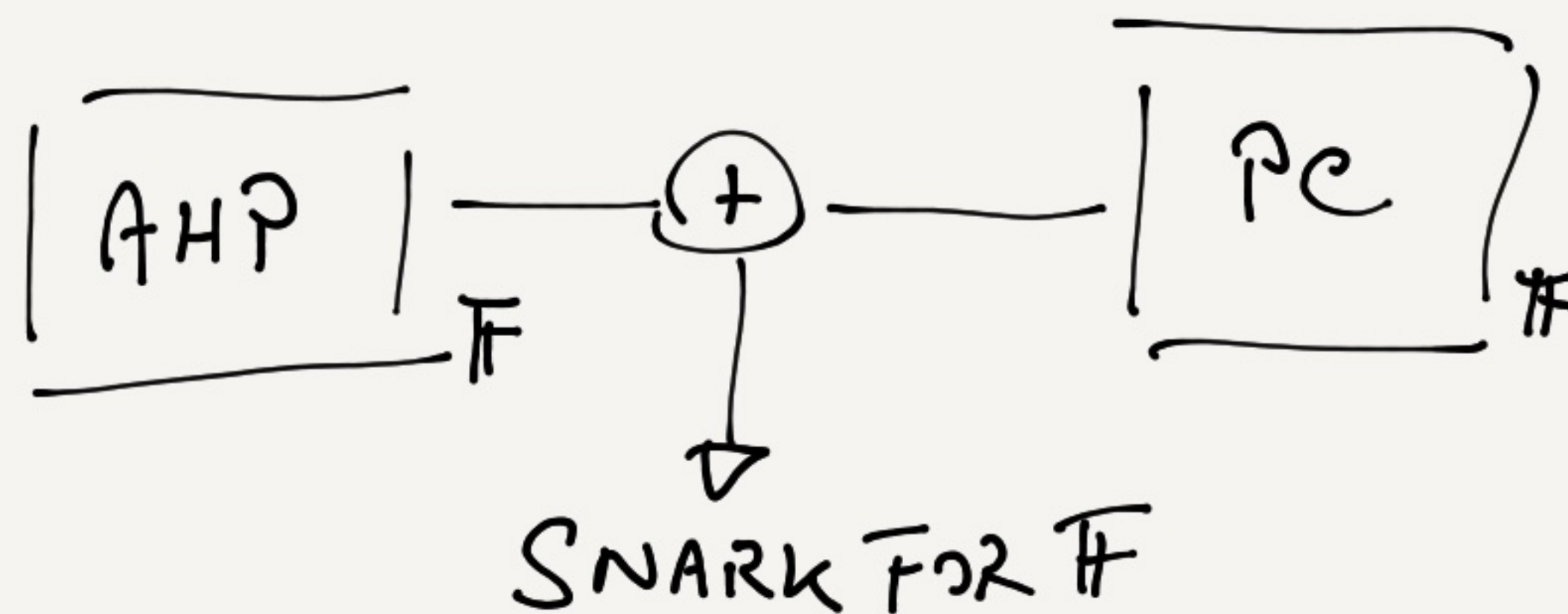
$$\left(\begin{array}{l} z(p, \dots) \cdot p_1(p, \dots) + \sigma p_2(p, \dots) \stackrel{?}{=} \dots \pmod{q} \\ p_1(\sigma, \dots) \cdot p_2(\sigma, \dots) \stackrel{?}{=} \dots \pmod{q} \end{array} \right)$$



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

\swarrow
FIRST MILESTONE TOWARDS
FULL-SUCCINCTNESS.

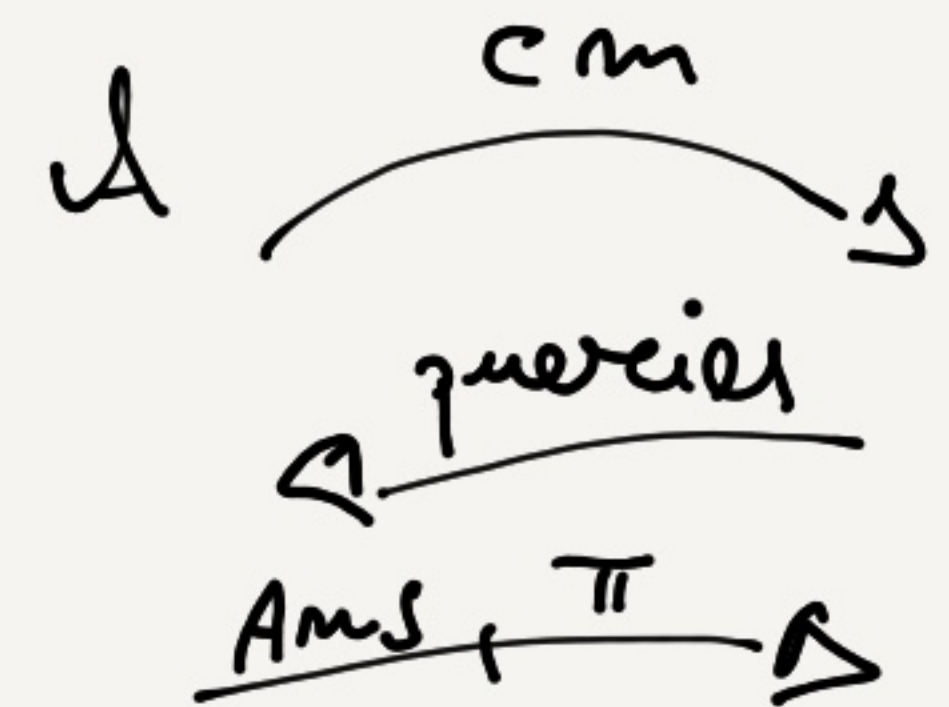
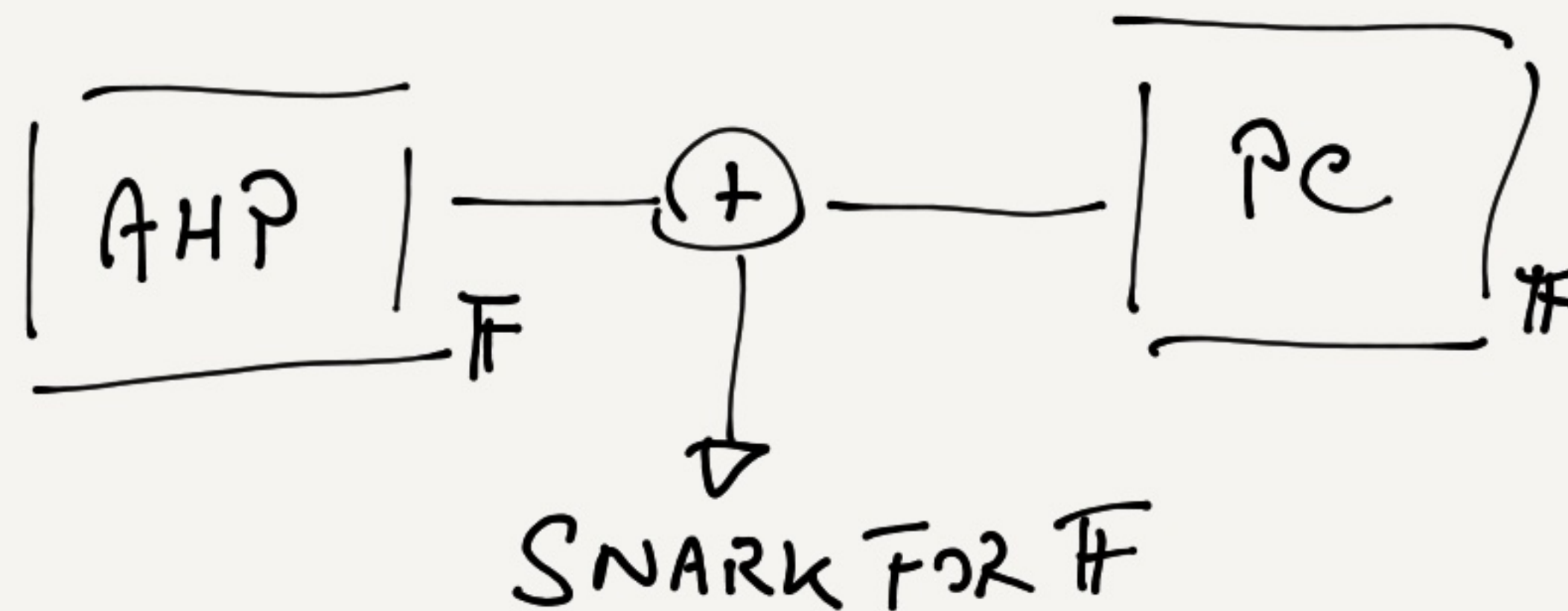
$A \text{ HP} + \text{PC} \xrightarrow{\text{fingerprint}} \text{mod-AHP} + ?$
 \nwarrow
 Poly Comm



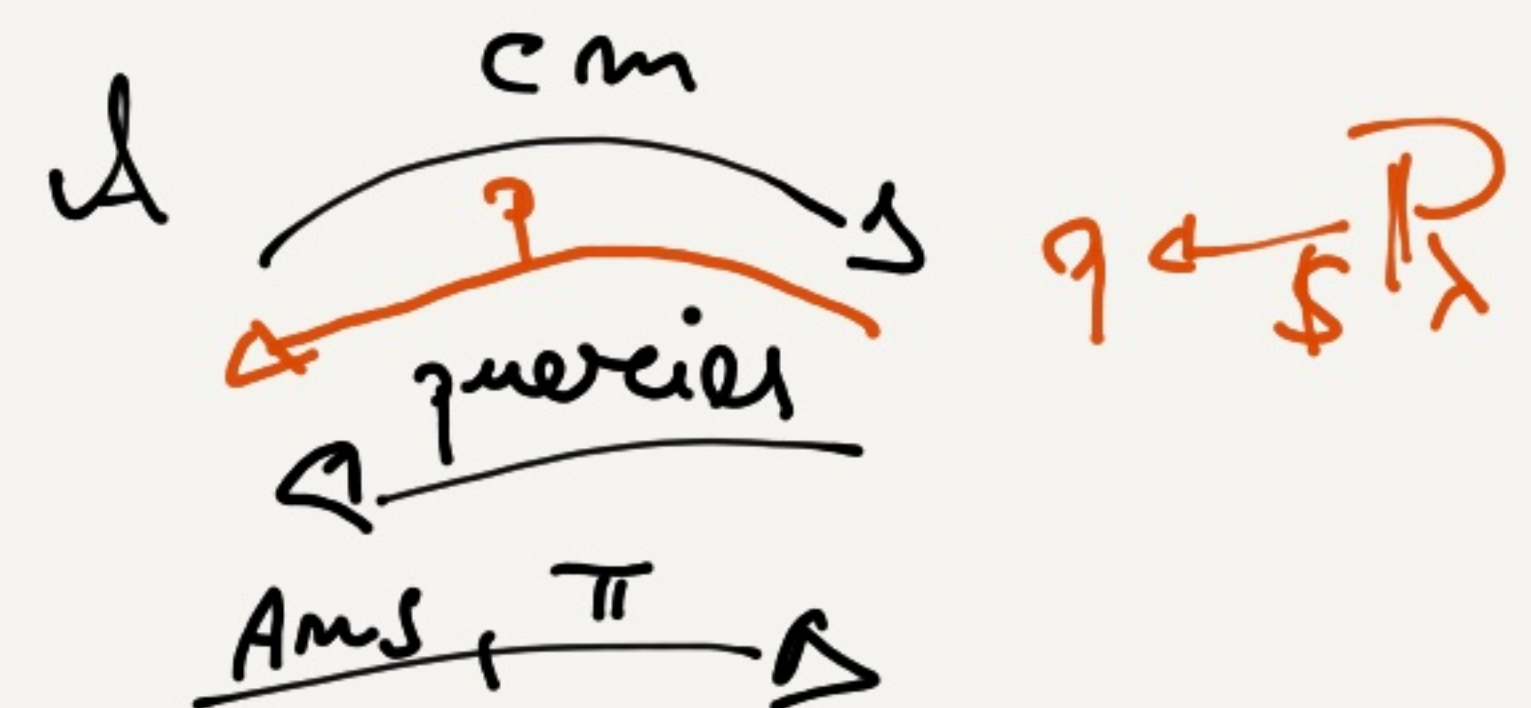
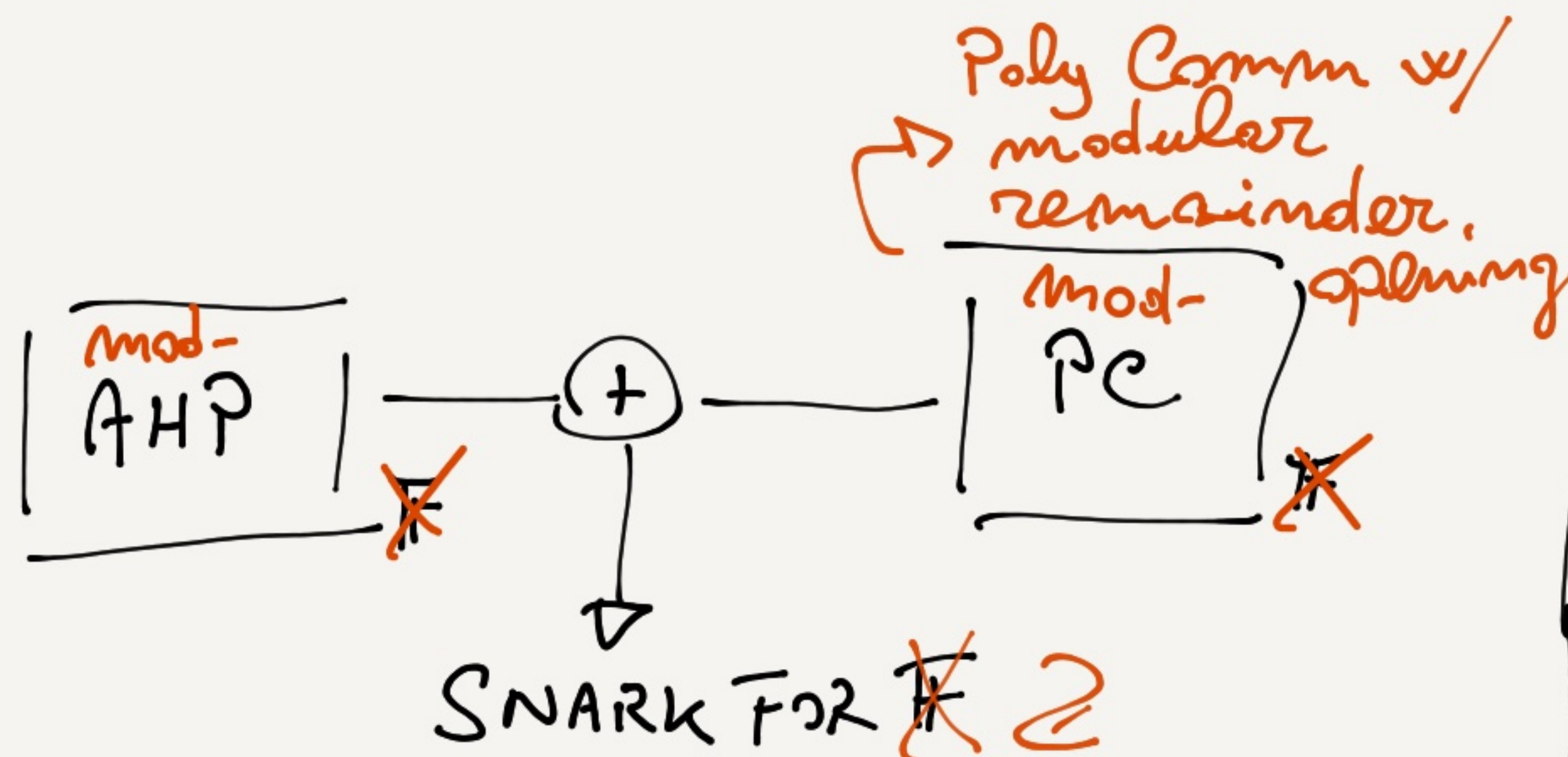
 $A \text{ HP} + \text{PC} \mapsto \text{mod-AHP} + ?$

\nwarrow
Poly Comm

STANDARD
CASE



Ext $\rightarrow g$:
 $open(cm) = g$
 $g(queries) = Ans$



Ext $\rightarrow g$:
 $open(cm) = g$
 $g(queries) = Ans$
 $(mod \ 9)$

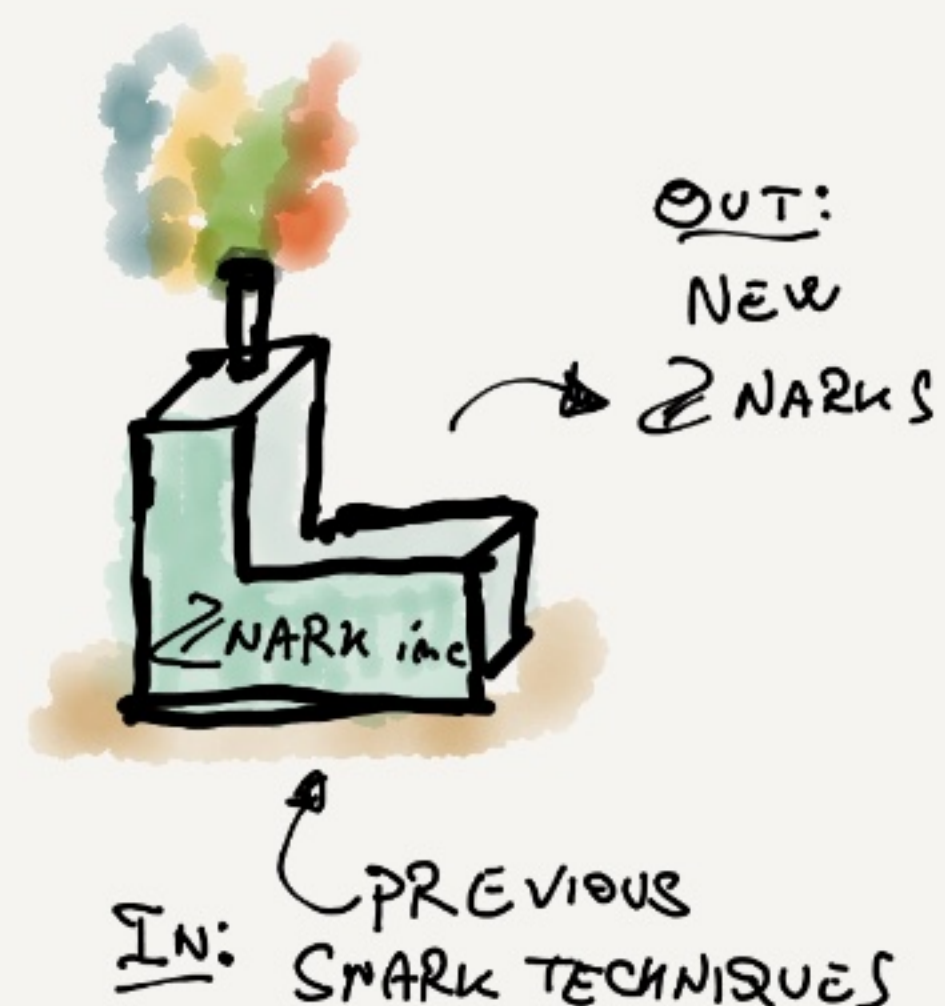
KSND IN mod-AHP

KSND AHP : $Ext \rightarrow \bar{x} \in \mathbb{F}^m$:
(STANDARD) $\mathcal{C}(\bar{x}) \checkmark$

NATURAL : $\bar{E} \times T \rightarrow \bar{x} \in \mathbb{Z}^m$
KSND mod-AHP $\mathcal{C}(\bar{x}) \checkmark$

KSNs IN mod-AHP

kSNs AHP : $E_{xT} \rightarrow \bar{x} \in \mathbb{F}^m$:
(STANDARD) $C(\bar{x}) \checkmark$



NATURAL : $\bar{E}_{xT} \rightarrow \bar{x} \in \mathbb{Z}^m$
 $C(\bar{x}) \checkmark$

IMPLIES

SIMPLER ("WEAK") :
kSNs mod-AHP

$E_{xT} \rightarrow \bar{x} \in \mathbb{F}_q^m \quad (q \leftarrow \mathbb{P}_\lambda)$
 $[C]_q(\bar{x}) \checkmark$

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$$[A]_{\bar{x}} \cdot [B]_{\bar{x}} - [C]_{\bar{x}} = 0$$

VARIANT
OF
SPARTAN
(SETTY '20)

$\mathcal{P}(\bar{x})$

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

$\checkmark \hat{A}, \hat{B}, \hat{C}$

$$q \leftarrow \mathcal{P}_r$$

OVER
 \mathbb{F}_q

SUMCHECK1

...

SUMCHECK2

$$\dots + \tilde{w}(p) + \dots \stackrel{?}{=} 0 \pmod{q}$$

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

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$$[A]_{\bar{w}} \cdot [B]_{\bar{w}} - [C]_{\bar{w}} = 0$$

VARIANT
OF
SPARTAN
(SETTY '20)

$P(\bar{w})$

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

$\checkmark \hat{A}, \hat{B}, \hat{C}$

$$q \leftarrow \mathbb{P}_r$$

OVER
 \mathbb{F}_q

SUMCHECK1

...

SUMCHECK2

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

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

EASY: PROVING

SIMPLER ("WEAK"):
KINDS mod-AHP

$$E_{x^T} \rightarrow \bar{w} \in \mathbb{F}_q^n \quad (q \leftarrow \mathbb{P}_r)$$

$$[C]_q(\bar{w}) \checkmark$$

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

$$[A]_{\bar{w}} \cdot [B]_{\bar{w}} - [C]_{\bar{w}} = 0$$

VARIANT
OF
SPARTAN
(SETTY '20)

$P(\bar{w})$

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

$\checkmark \tilde{A}, \tilde{B}, \tilde{C}$

$$q \leftarrow \mathbb{P}_r$$

OVER
 \mathbb{F}_q

SUMCHECK 1

...

SUMCHECK 2

$$\dots + \tilde{w}(p) + \dots \stackrel{?}{=} 0 \pmod{q}$$

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

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

EASY: PROVING

SIMPLER ("WEAK"):
KINDS mod-AHP

$$Ext \rightarrow \bar{w} \in \mathbb{F}_q^m \quad (q \leftarrow \mathbb{P}_r)$$

$$[C]_q(\bar{w}) \checkmark$$

NOT EASY:

MAKING \checkmark

NOT PAY FOR $\tilde{A}, \tilde{B}, \tilde{C}$

SPARTAN'S
DENSE 2 SPARSE
COMPILER

EXTENSIONS TO
FRAMEWORK
("DELAYED INPUT" mod-AHPs.)
STRONGER mod-PC

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

INSTANTIATIONS (AND THEIR CHALLENGES): mod-AHP

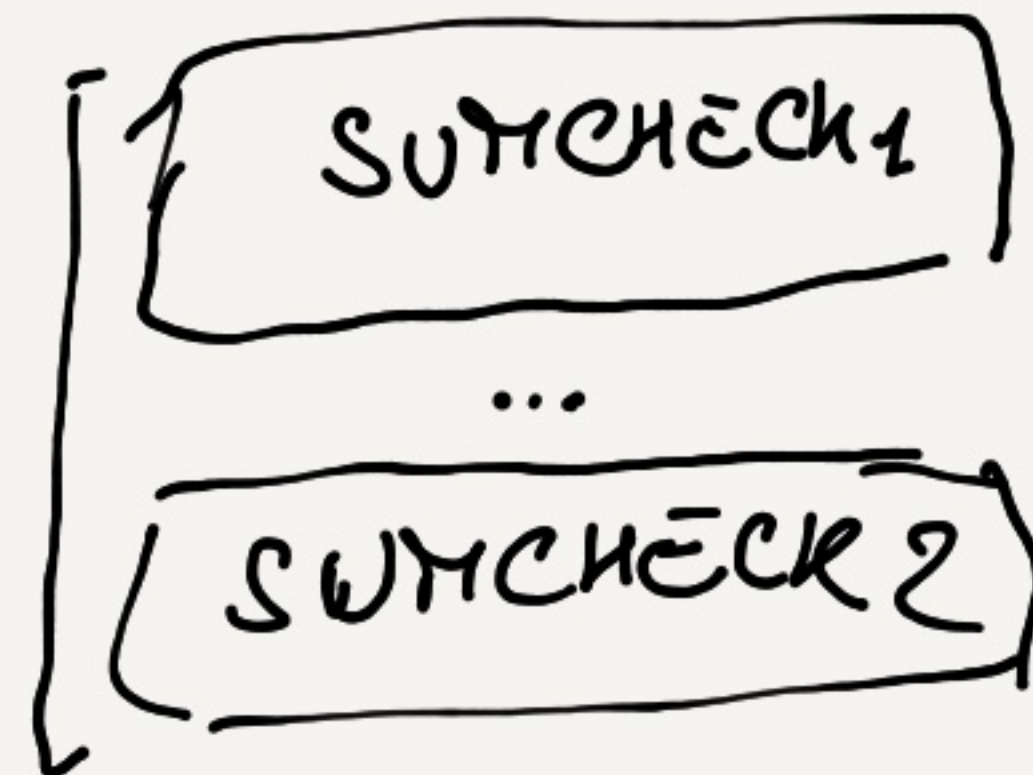
$$[A]_{\bar{w}} \cdot [B]_{\bar{w}} - [C]_{\bar{w}} = 0$$

VARIANT
OF
SPARTAN
(SETTY '20)

$P(\bar{w})$

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

OVER
 \mathbb{F}_q



$\checkmark \hat{A}, \hat{B}, \hat{C}$

$$q \leftarrow \mathbb{P}_r$$

$$\dots + \tilde{w}(p) + \dots \stackrel{?}{=} 0 \pmod{q}$$

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

EFFICIENCY FEATURES
(FROM SPARTAN):

- LINEAR
- VERY PARALLELIZABLE (SIMD)

EXAMPLE: FOR RSA SIGNATURES ($|\bar{w}| = 16$)
YOU ARE RUNNING
SPARTAN ON A SIZE 16 CIRCUIT

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

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

Q: LARGE INTEGER

INSTANTIATIONS (AND THEIR CHALLENGES) : mod-PC

OUR STARTING
POINT

TECHNIQUES FROM
GROUPS OF UNKNOWN
ORDER

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

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

Q: LARGE INTEGER

CHALLENGE 1:

BINDING OF DARK
INSUFFICIENT

DOES NOT RECOVER $f(\hat{x})$
(A POLY)

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

INSTANTIATIONS (AND THEIR CHALLENGES) : mod-PC

OUR STARTING
POINT

TECHNIQUES FROM
GROUPS OF UNKNOWN
ORDER

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

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

Q: LARGE INTEGER

CHALLENGE 1: BINDING OF DARK
INSUFFICIENT

USE Block et al.
INSTEAD AS MAIN TOOL

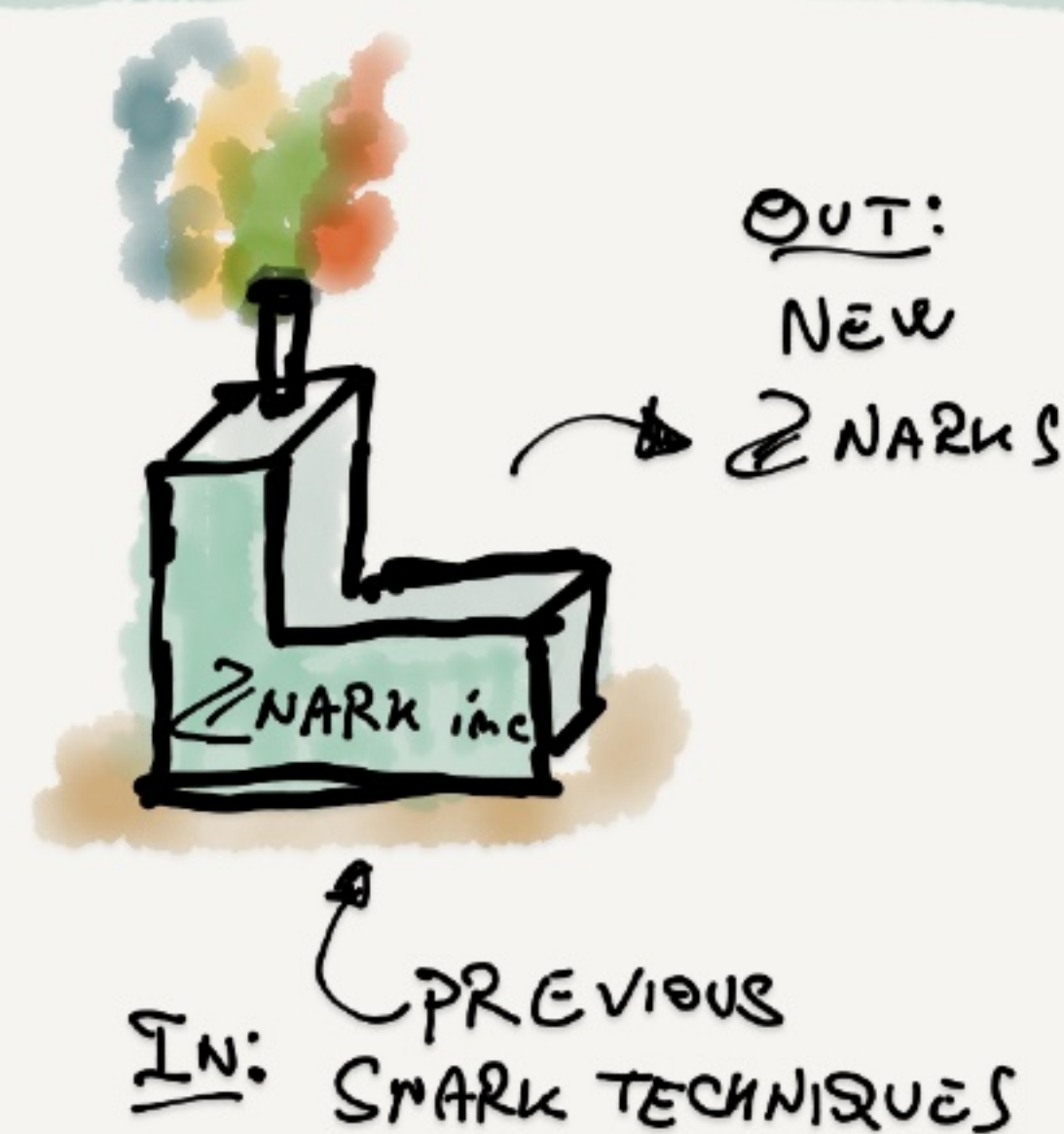
CHALLENGE 2: NOT SUCCINCT
IN $\|\bar{w}\|_\infty$

DOES NOT RECOVER $f(\bar{x})$
(A POLY)
BUT $\frac{f(\bar{x})}{N}$ (A RATIONAL
FUNCTION)

REQUIRE ADDITIONAL
BUILDING BLOCKS
(APPROPRIATE ARC 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!