One Framework, Many Assumptions: Modern MPC-in-the-Head Signatures

Thibauld Feneuil

Workshop AM-PQC

August 13, 2026, *Ohrid*

Introduction

Signature size
(in kilobytes)

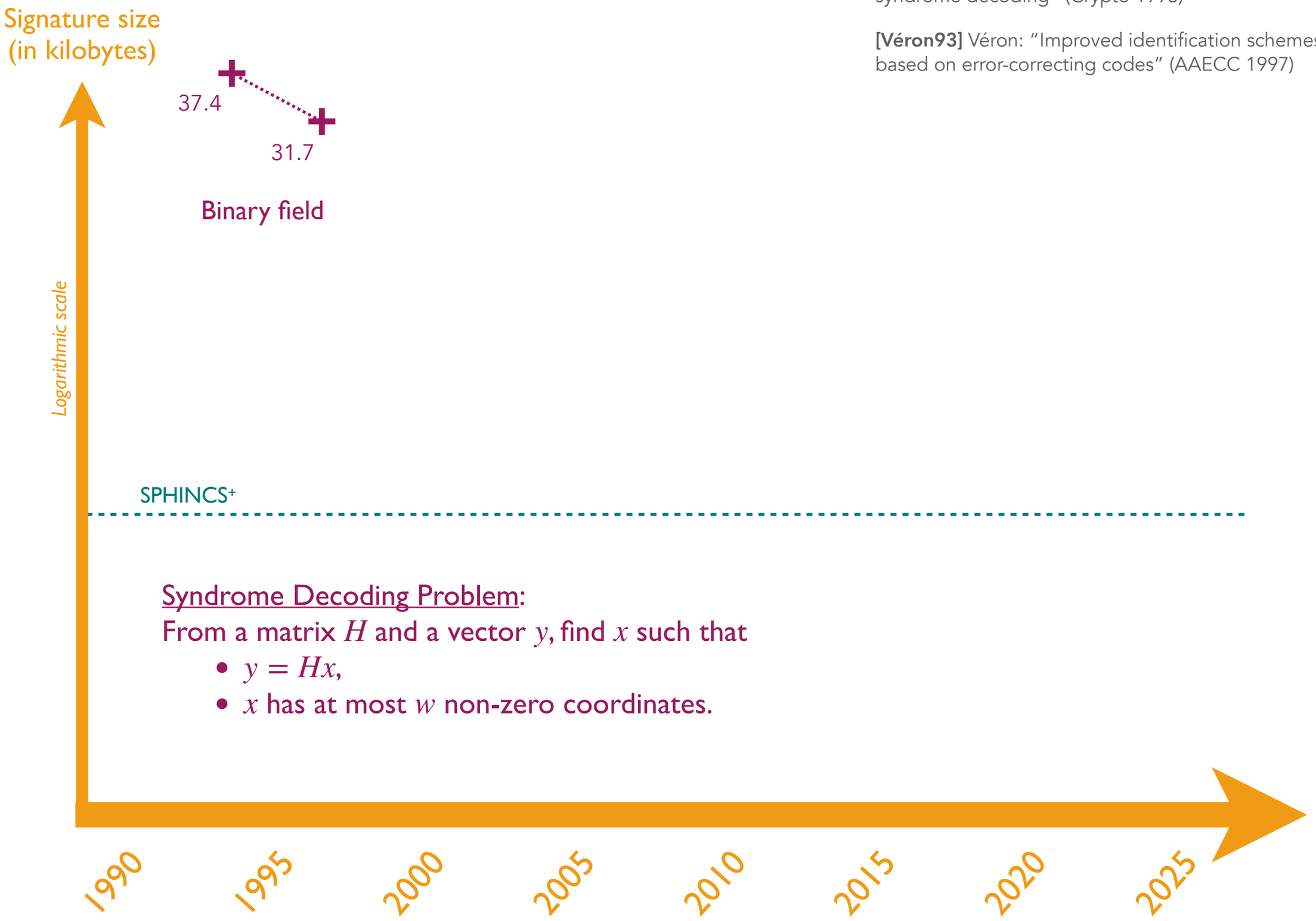
Logarithmic scale

SPHINCS+

Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

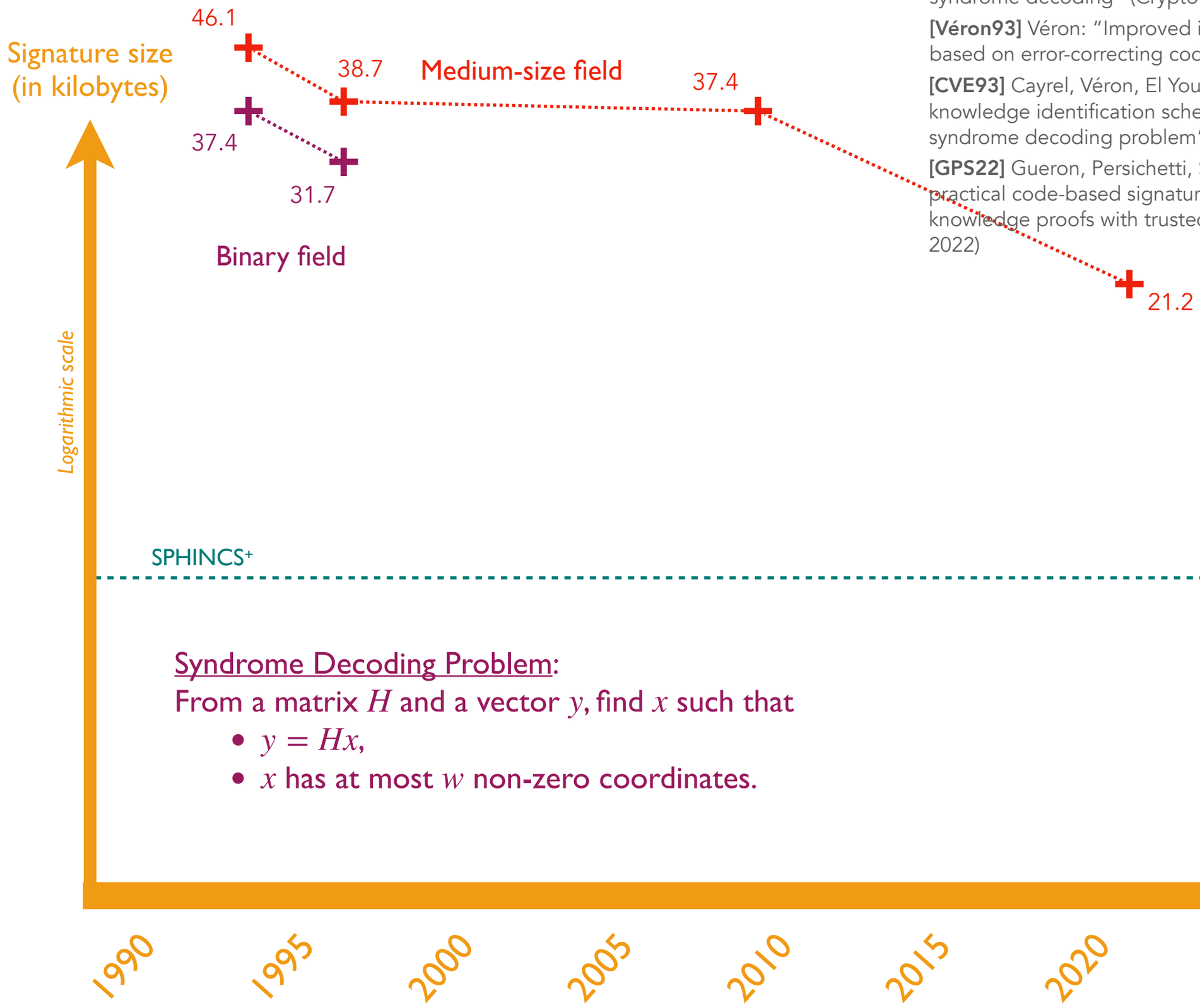
- $y = Hx$,
- x has at most w non-zero coordinates.

1990 1995 2000 2005 2010 2015 2020 2025



[Stern93] Stern: "A new identification scheme based on syndrome decoding" (Crypto 1993)

[Véron93] Véron: "Improved identification schemes based on error-correcting codes" (AAECC 1997)

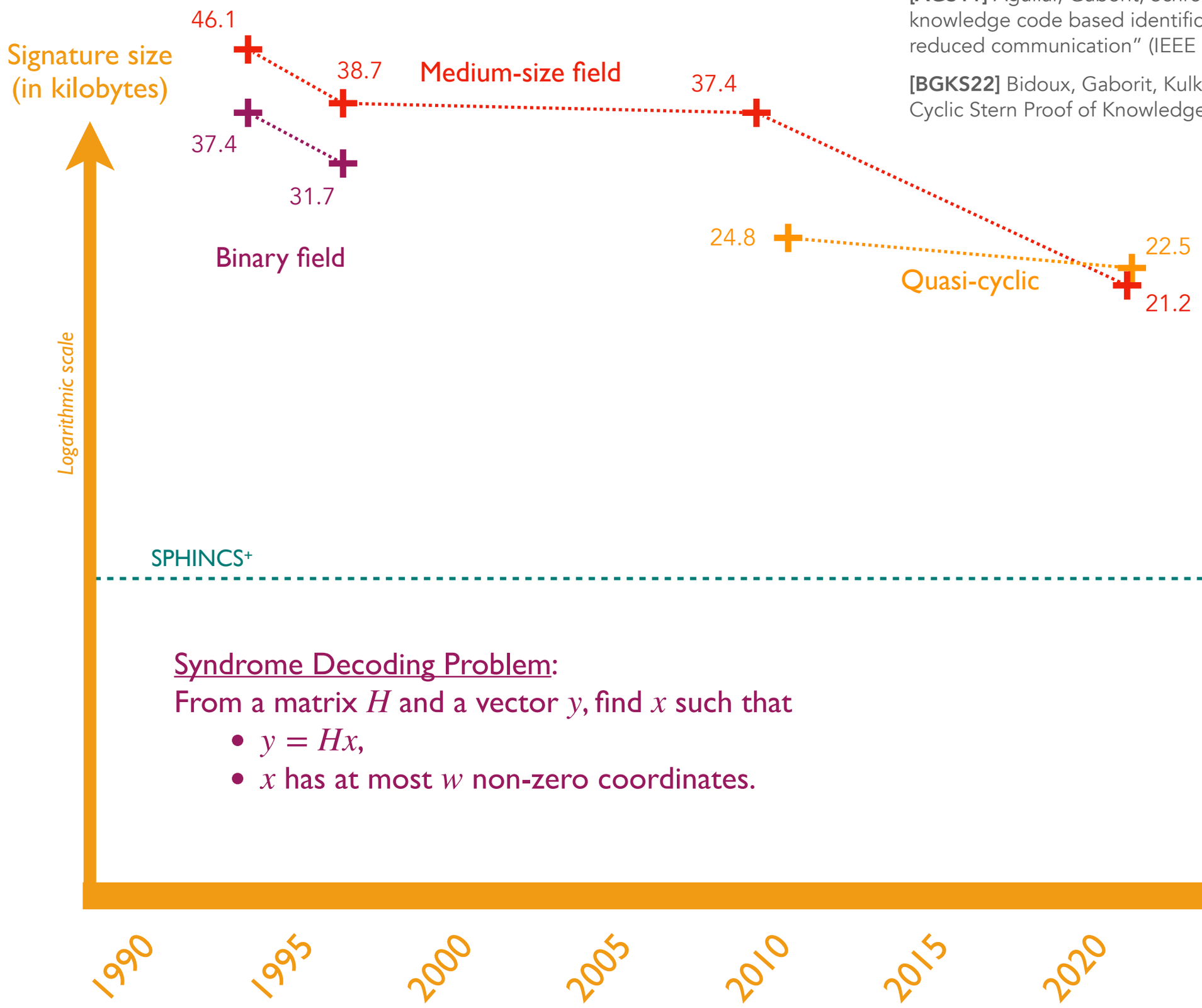


[Stern93] Stern: "A new identification scheme based on syndrome decoding" (Crypto 1993)

[Véron93] Véron: "Improved identification schemes based on error-correcting codes" (AAECC 1997)

[CVE93] Cayrel, Véron, El Yousfi Alaoui: "A zero-knowledge identification scheme based on the q-ary syndrome decoding problem" (SAC 2010)

[GPS22] Gueron, Persichetti, Santini: "Designing a practical code-based signature scheme from zero-knowledge proofs with trusted setup" (Cryptography 2022)



[AGS11] Aguilar, Gaborit, Schrek: "A new zero-knowledge code based identification scheme with reduced communication" (IEEE ITW 2011)

[BGKS22] Bidoux, Gaborit, Kulkarni, Sendrier: "Quasi-Cyclic Stern Proof of Knowledge" (ISIT 2022)

[IKOS07] Ishai, Kushilevitz, Ostrovsky, Sahai: "Zero-knowledge from secure multiparty computation" (STOC 2007)

Signature size (in kilobytes)

Logarithmic scale

46.1

38.7

Medium-size field

37.4

37.4

31.7

Binary field

24.8

22.5

Quasi-cyclic

21.2

IKOS07

SPHINCS+

Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

1990 1995 2000 2005 2010 2015 2020 2025

[FJR23] Feneuil, Joux, Rivain: "Shared Permutation for Syndrome Decoding: New Zero-Knowledge Protocol and Code-Based Signature" (ePrint 2021/1576, Journal DCC)

Signature size (in kilobytes)

Logarithmic scale

46.1

38.7

Medium-size field

37.4

37.4

31.7

Binary field

24.8

Quasi-cyclic

22.5

21.2

16.0

IKOS07

SPHINCS+

Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

1990 1995 2000 2005 2010 2015 2020 2025

👉 Shared-permutation framework

[FJR22] Feneuil, Joux, Rivain: "Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs" (Crypto 2022)

Signature size (in kilobytes)

Logarithmic scale

46.1

38.7

Medium-size field

37.4

37.4

31.7

Binary field

24.8

Quasi-cyclic

22.5

21.2

16.0

12.1

SDitH v1

8.5

IKOS07

SPHINCS+

Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

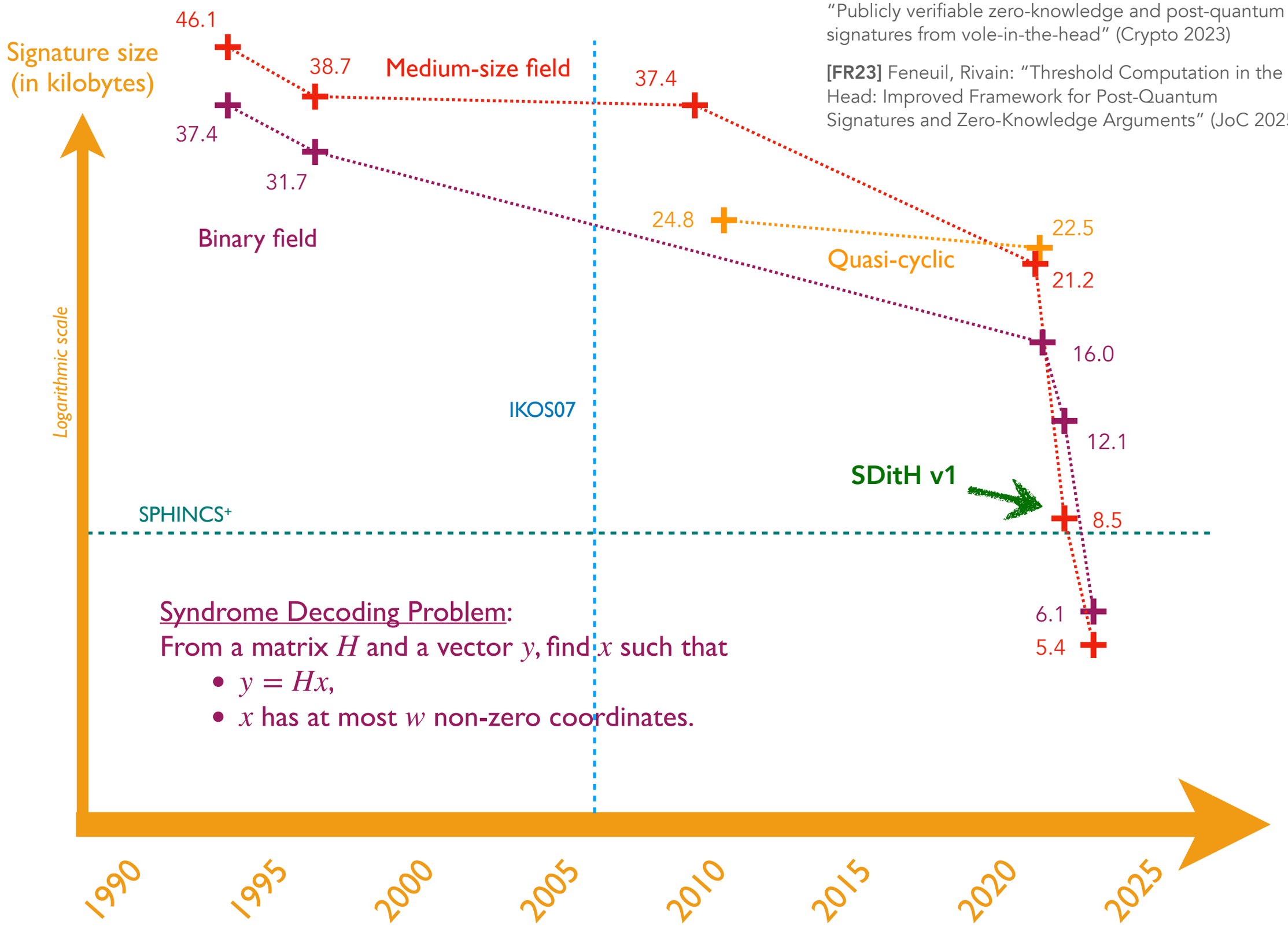
- $y = Hx$,
- x has at most w non-zero coordinates.

1990 1995 2000 2005 2010 2015 2020 2025

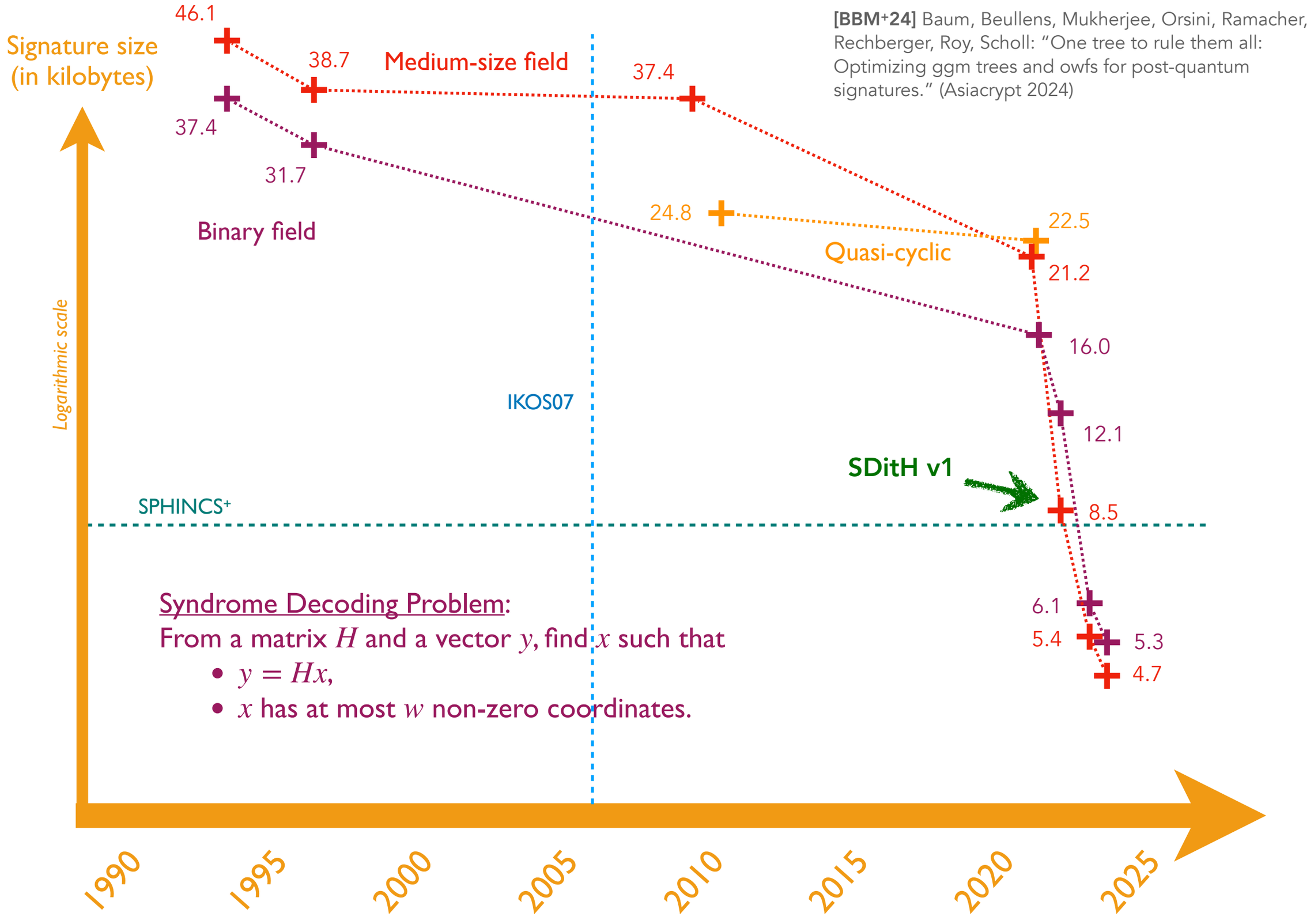
👉 Additive-sharing-based framework

[BBD+23] Baum, Braun, Delpech, Klooß, Orsini, Roy, Scholl: "Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head" (Crypto 2023)

[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (JoC 2025)



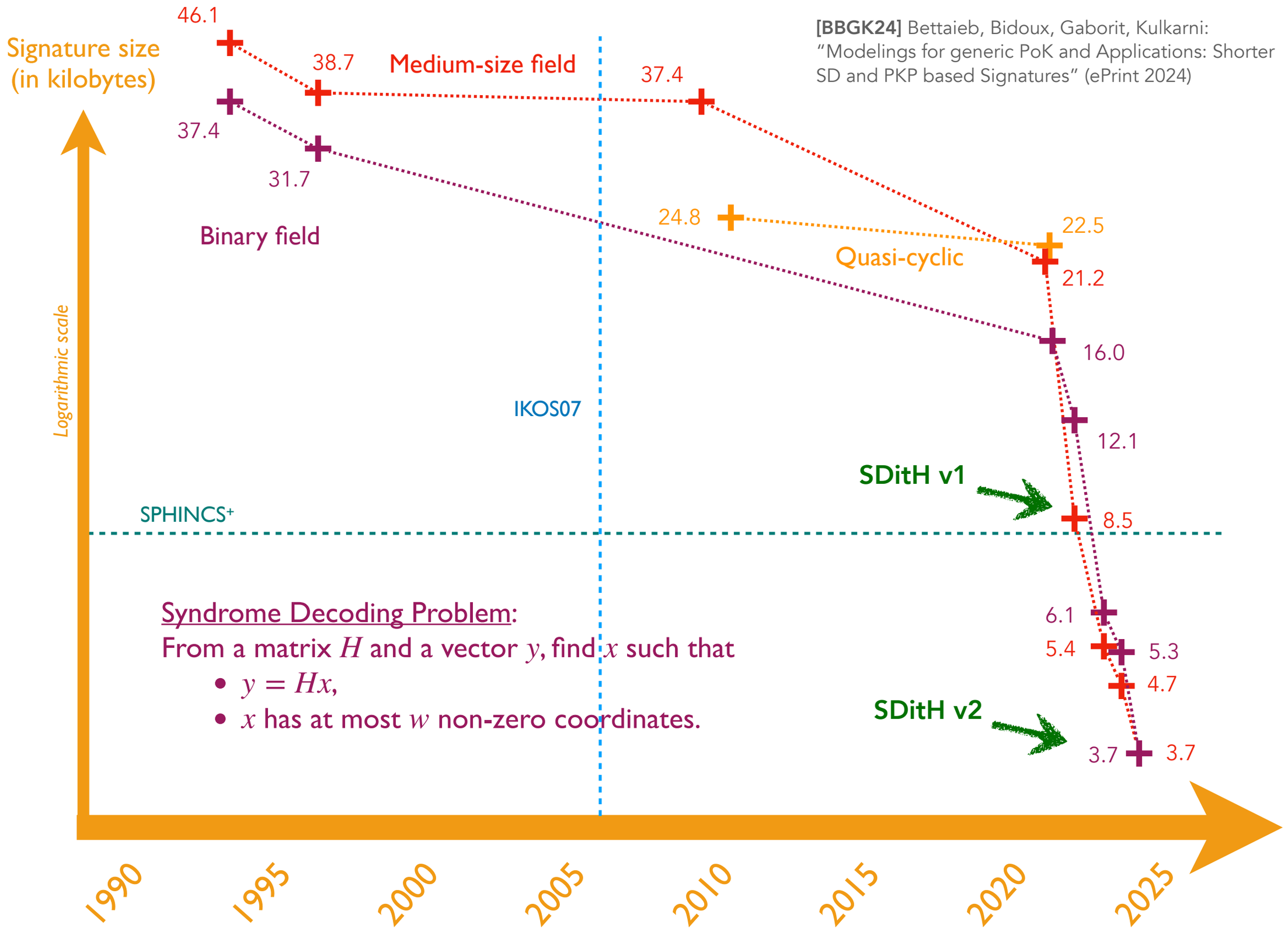
👉 PIOP-based frameworks (VOLEitH, TCitH)



[BBM+24] Baum, Beullens, Mukherjee, Orsini, Ramacher, Rechberger, Roy, Scholl: "One tree to rule them all: Optimizing ggm trees and owfs for post-quantum signatures." (Asiacrypt 2024)

👉 Combinatorial optimizations

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni: "Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures" (ePrint 2024)



Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

👉 New modeling of the SD problem

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni: "Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures" (ePrint 2024)

Signature size (in kilobytes)

Logarithmic scale

46.1

38.7

Medium-size field

37.4

37.4

31.7

Binary field

24.8

Quasi-cyclic

22.5

21.2

16.0

12.1

8.5

6.1

5.4

5.3

4.7

3.7

3.7

MPC-in-the-Head

IKOS07

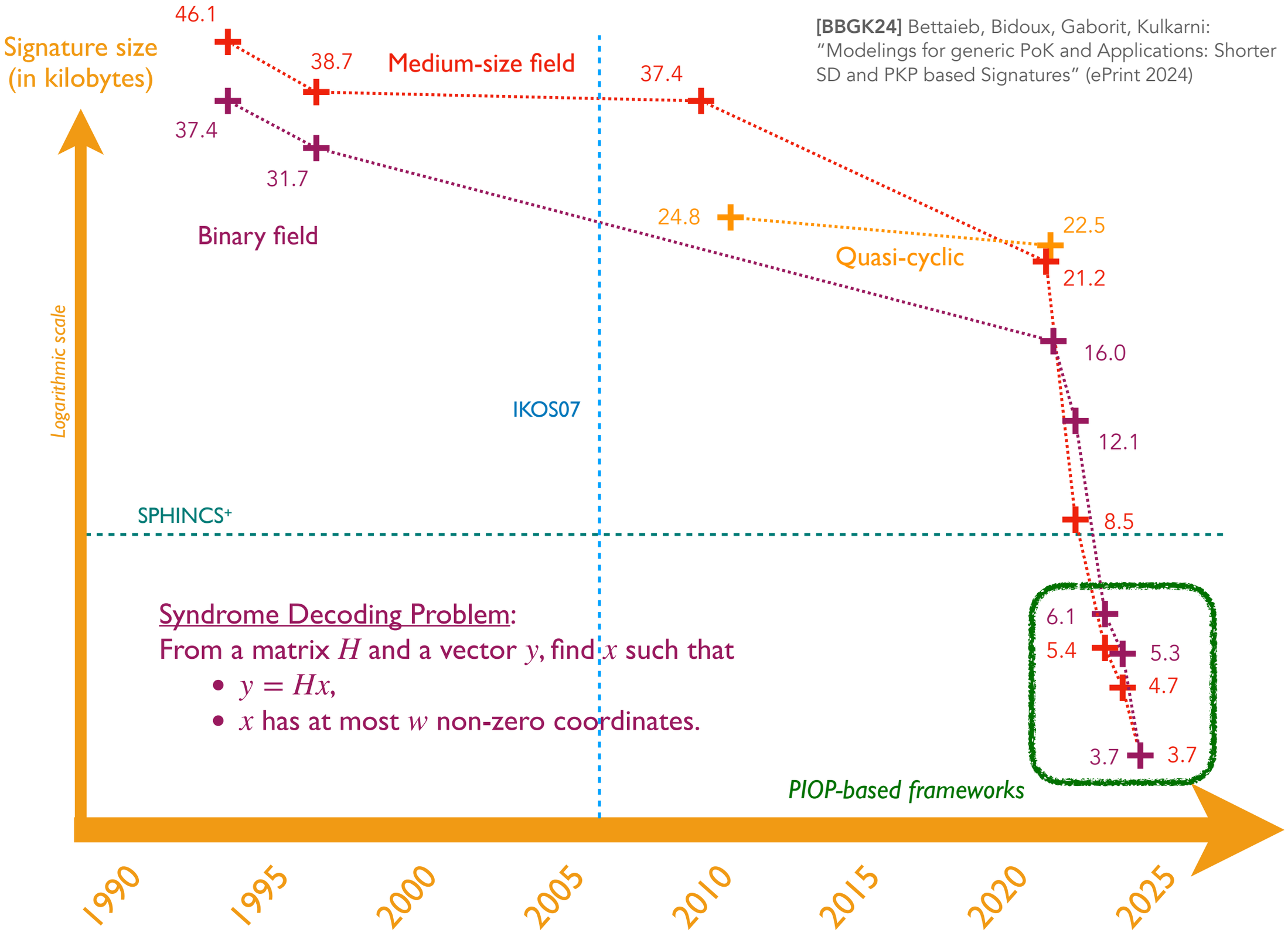
SPHINCS+

Syndrome Decoding Problem:

From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

1990 1995 2000 2005 2010 2015 2020 2025



[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni: "Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures" (ePrint 2024)

Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

For the sake of diversity…

AES Key Recovery
AIM Key Recovery
LowMC Key Recovery
Rain Key Recovery

Anemoi Hash Preimage
Poseidon Hash Preimage
Griffin Hash Preimage
RescuePrime Hash Preimage

Legendre PRF
BHHG's PRF

LWE
SIS
Subset Sum

Discrete Logarithm
Integer Factorization
Double Discrete Logarithm

Syndrome Decoding
Regular Syndrome Decoding
Permuted Kernel
Linear Code Equivalence
Restricted Syndrome Decoding

MinRank
Rank Syndrome Decoding
Subfield Collision Problem
Matrix Subcode Equivalence

Multivariate Quadratic
PowAff2

For the sake of diversity…

AES Key Recovery
AIM Key Recovery
LowMC Key Recovery
Rain Key Recovery

Anemoi Hash Preimage
Poseidon Hash Preimage
Griffin Hash Preimage
RescuePrime Hash Preimage

LWE
SIS
Subset Sum

Syndrome Decoding
Regular Syndrome Decoding
Permuted Kernel
Linear Code Equivalence
Restricted Syndrome Decoding

MinRank
Rank Syndrome Decoding
Subfield Collision Problem
Matrix Subcode Equivalence

Multivariate Quadratic
PowAff2

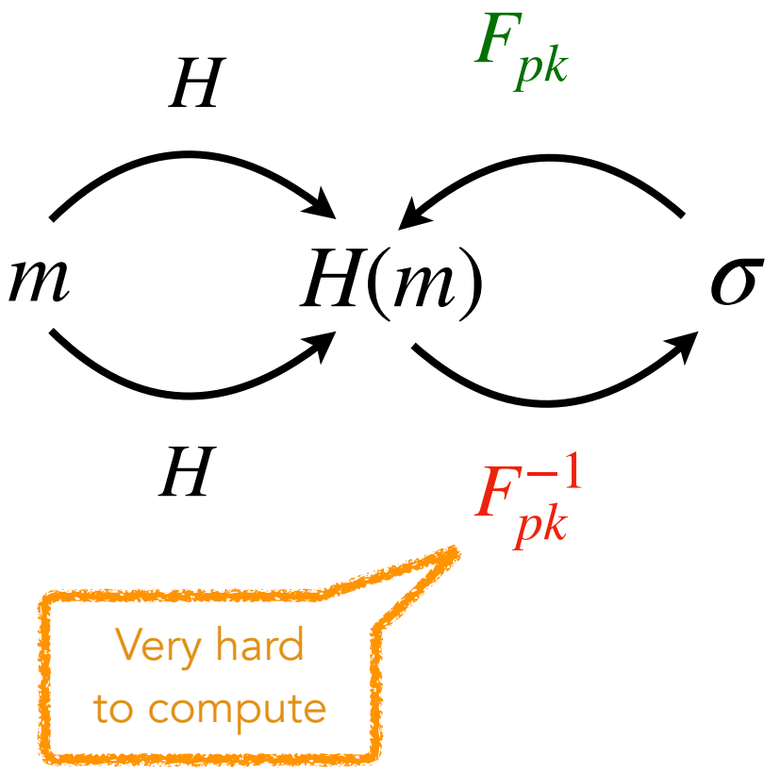
Legendre PRF
BHHG's PRF

Discrete Logarithm
Integer Factorization
Double Discrete Logarithm

We can build highly conservative schemes!

How to build signature schemes?

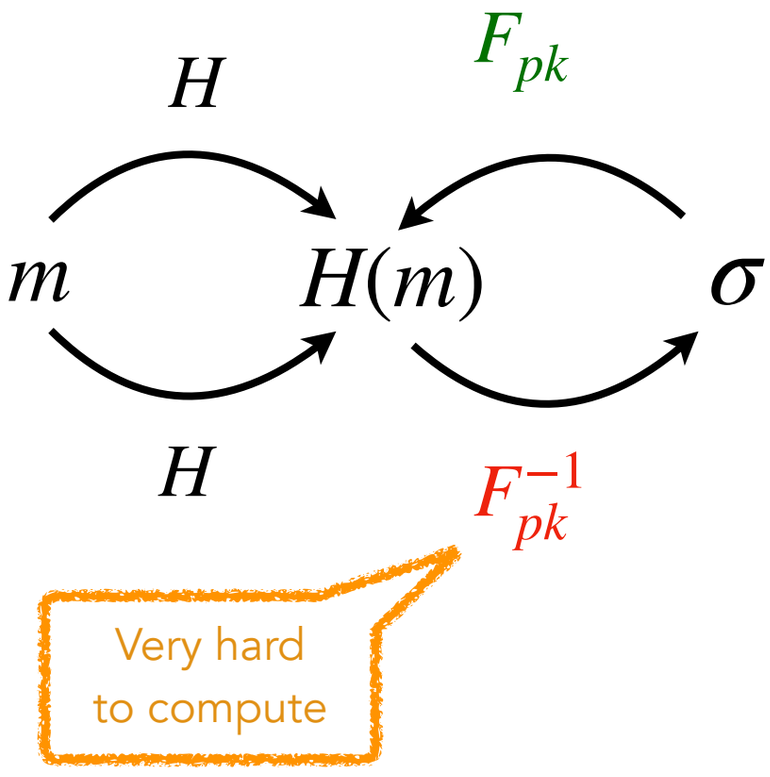
Hash & Sign



- Short signatures
- “Trapdoor” in the public key

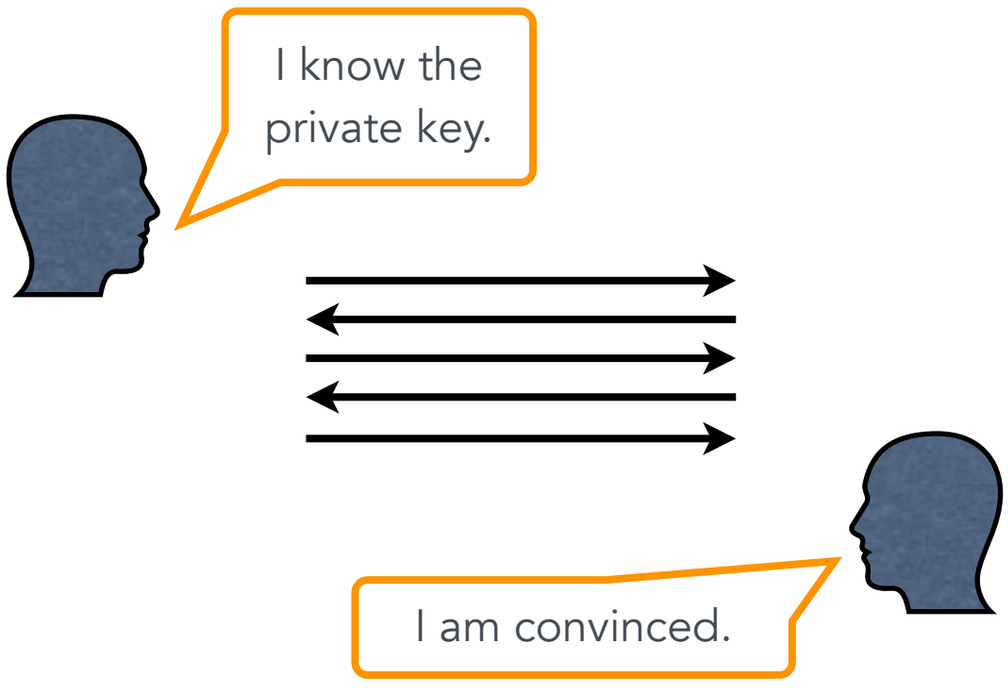
How to build signature schemes?

Hash & Sign



- Short signatures
- "Trapdoor" in the public key

From an identification scheme



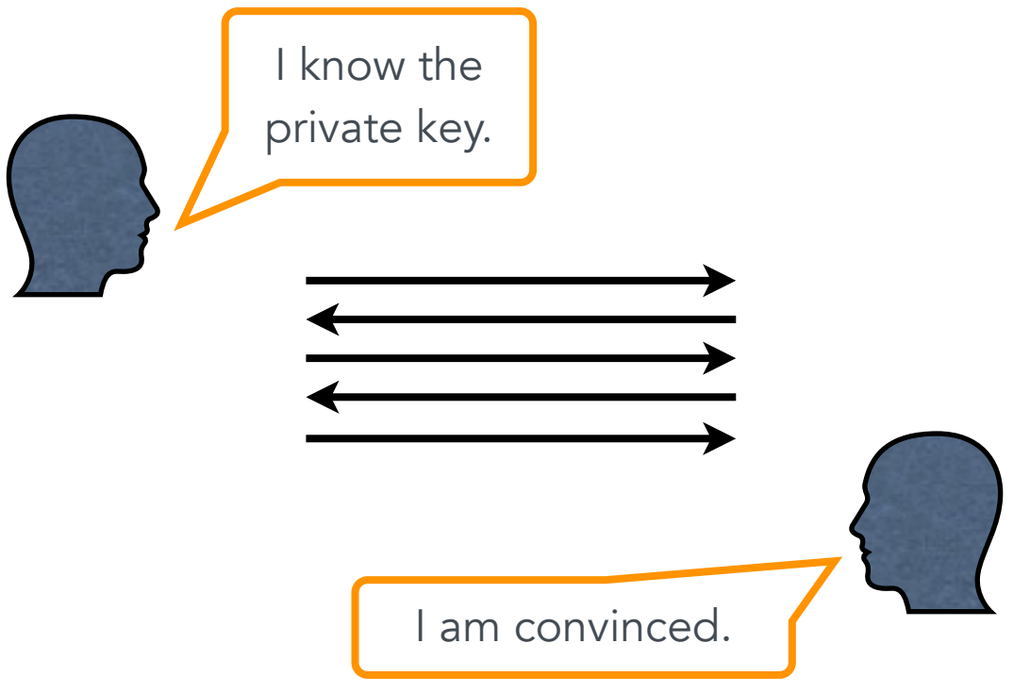
- Large(r) signatures
- Short public key

How to build signature schemes?

Hash & Sign

- Short signatures
- "Trapdoor" in the public key

From an identification scheme

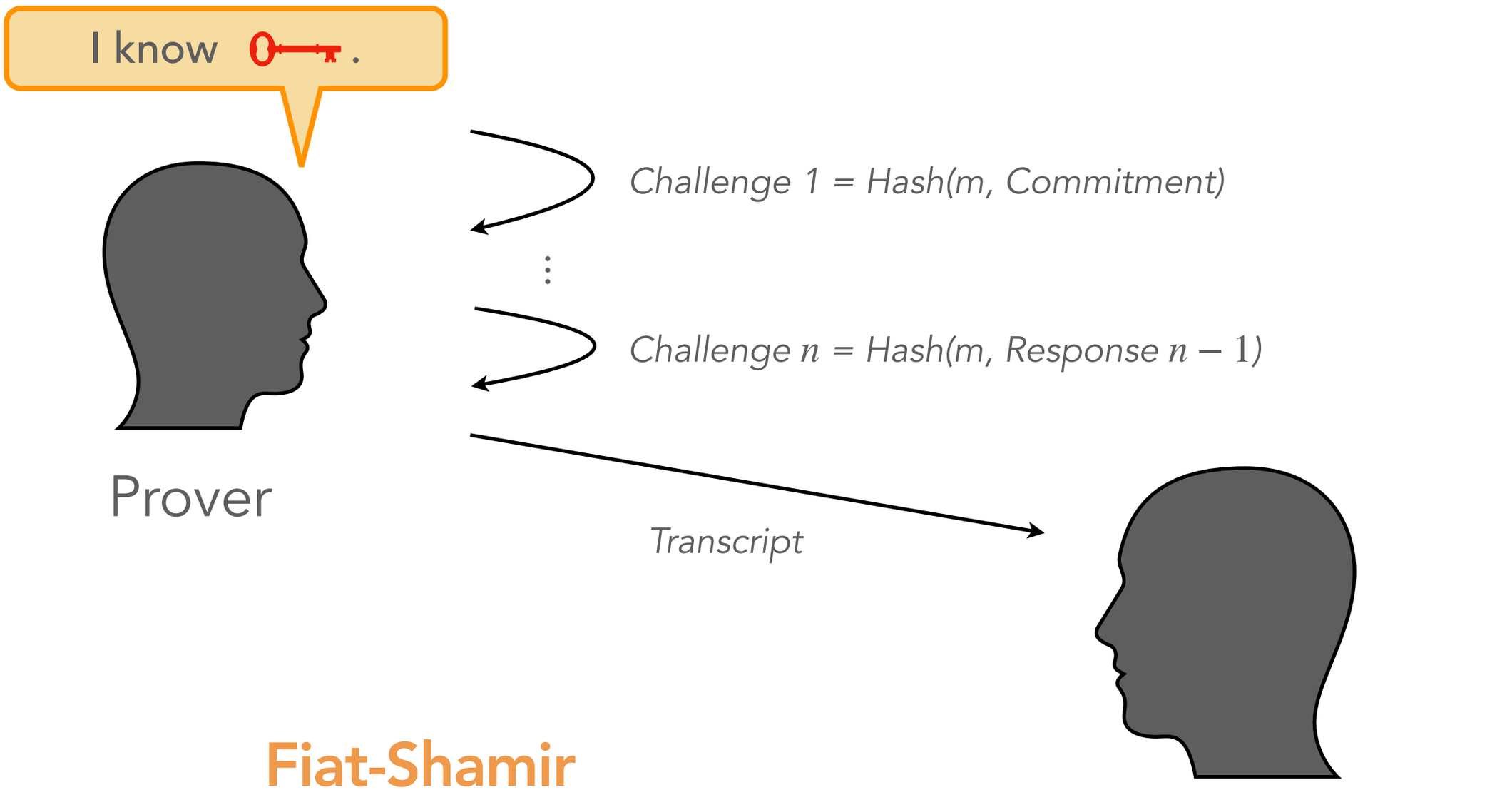


- Large(r) signatures
- Short public key

Identification Scheme

- **Completeness:** Pr[verif ✓ | honest prover] = 1
- **Soundness:** Pr[verif ✓ | malicious prover] $\leq \varepsilon$ (e.g. 2^{-128})
- **Zero-knowledge:** verifier learns nothing on .

Identification Scheme



Fiat-Shamir Transformation

m: message to sign

Polynomial Interactive Oracle Proof

PIOP-based MPCitH Frameworks

[BBD+23] Baum, Braun, Delpech, Klooß, Orsini, Roy, Scholl. Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head. Crypto 2023.

[FR25] Feneuil, Rivain. *Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments*. Journal of Cryptology, 2025.

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

I know $w_1, \ldots, w_n$ such that

$$f(w_1, \ldots, w_n) = 0$$

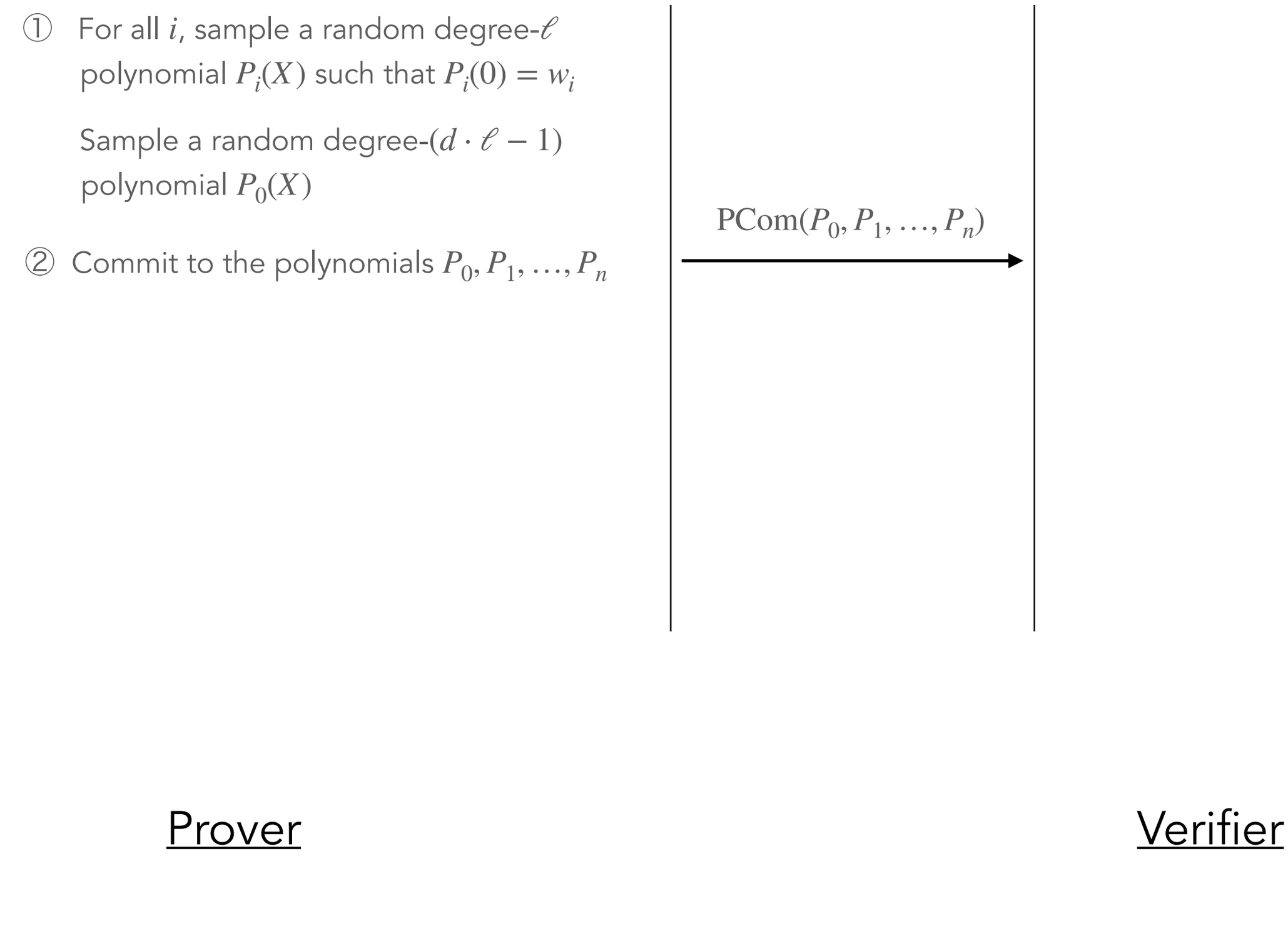
where f is a public **degree-d polynomial**.

Prover

Prove it!

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*



TCitH and VOLEitH Frameworks, *in the PIOP formalism*

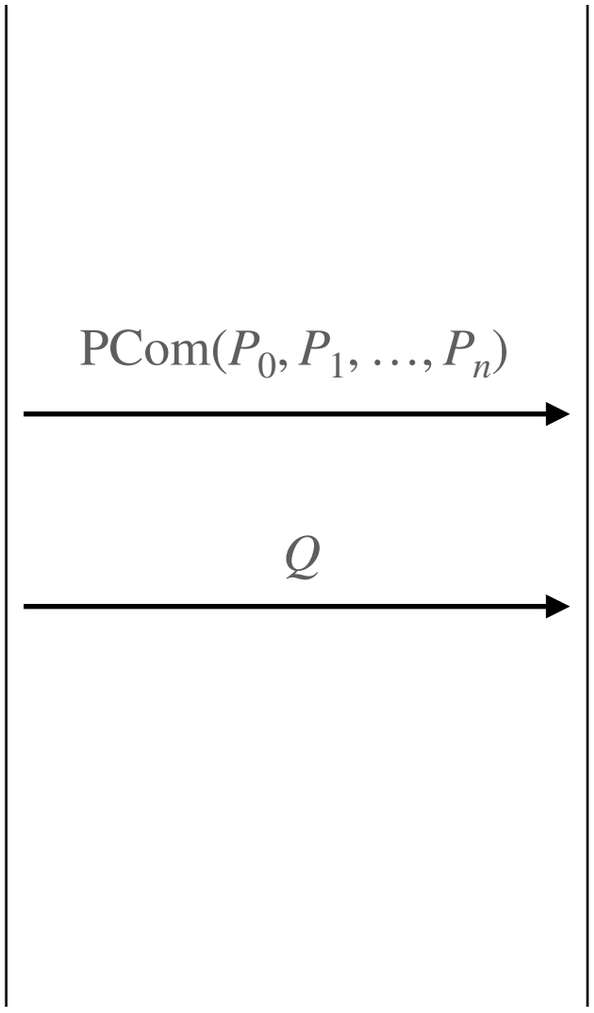
① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$



Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomial $Q(X)$ such that

$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$

Q →

Well-defined!

$0 \cdot P_0(0) + f(P_1(0), \ldots, P_n(0)) = 0 + f(w_1, \ldots, w_n) = 0$

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q →

← $r_1, \ldots, r_\ell$

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Soundness Analysis

① For all i, choose a degree-ℓ polynomial $P_i(X)$. We have

$$f(P_1(0), \ldots, P_n(0)) \neq 0.$$

Choose a degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

③ Reveal a polynomial $Q(X)$. We know that

$$X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

Q →

← $r_1, \ldots, r_\ell$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Malicious Prover 😈

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Soundness Analysis

① For all i, choose a degree-ℓ polynomial $P_i(X)$. We have

$$f(P_1(0), \ldots, P_n(0)) \neq 0.$$

Choose a degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

③ Reveal a polynomial $Q(X)$. We know that

$$X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

Evaluation into 0

$= 0$ $\neq 0$

$\text{PCom}(P_0, P_1, \ldots, P_n)$

Q

$r_1, \ldots, r_\ell$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Malicious Prover 😈

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Soundness Analysis

① For all i, choose a degree-ℓ polynomial $P_i(X)$. We have

$$f(P_1(0), \ldots, P_n(0)) \neq 0.$$

Choose a degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Reveal a polynomial $Q(X)$. We know that

$$X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Malicious Prover 😈

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Soundness Analysis

① For all i, choose a degree-ℓ polynomial $P_i(X)$. We have

$$f(P_1(0), \ldots, P_n(0)) \neq 0.$$

Choose a degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$

③ Reveal a polynomial $Q(X)$. We know that

$$X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q

$r_1, \ldots, r_\ell$

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Verifier

Schwartz-Zippel Lemma: Let D be the **non-zero** degree-$(d \cdot \ell)$ polynomial defined as

$$D := X \cdot Q(X) - X \cdot P_0(X) - f(P_1(X), \ldots, P_n(X))$$

We have

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, choose a degree-ℓ polynomial $P_i(X)$. We have

$$f(P_1(0), \ldots, P_n(0)) \neq 0.$$

Choose a degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$

③ Reveal a polynomial $Q(X)$. We know that

$$X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q

$r_1, \ldots, r_\ell$

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Verifier

Soundness Analysis

Schwartz-Zippel Lemma: Let D be the **non-zero** degree-$(d \cdot \ell)$ polynomial defined as

$$D := X \cdot Q(X) - X \cdot P_0(X) - f(P_1(X), \ldots, P_n(X))$$

We have

$$\Pr[\text{verification passes}] = \Pr\left[\forall j, D(r_j) = 0 \mid \{r_j\}_j \subset_\$ \mathscr{C}\right] \leq \frac{\binom{d \cdot \ell}{\ell}}{\binom{|\mathscr{C}|}{\ell}}.$$

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Zero-Knowledge Analysis

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Prover

Verifier 👀

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Revealing ℓ evaluations of $P_i(X)$ leaks no information about w_i.

Zero-Knowledge Analysis

Verifier 👀

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Zero-Knowledge Analysis

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$

③ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$$

Q

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$v_{j,i} = P_i(r_j)$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

⑥ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + f(v_{j,1}, \ldots, v_{j,n})$$

Revealing $Q(X)$ leaks no information about w_i, thanks to $P_0(X)$.

Verifier 👀

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

I know $w_1, \ldots, w_n$ such that

$$f(w_1, \ldots, w_n) = 0$$

where f is a public **degree-d polynomial**.

Prover

Prove it!

Verifier

Soundness Error = $\dfrac{\binom{d \cdot \ell}{\ell}}{\binom{|\mathscr{C}|}{\ell}}$

Probability that a malicious prover can convince the verifier.

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

I know $w_1, \ldots, w_n$ such that

$$\begin{cases} f_1(w_1, \ldots, w_n) & = 0 \\ & \vdots \\ f_m(w_1, \ldots, w_n) & = 0, \end{cases}$$

where $f_1, \ldots, f_m$ are public **degree-d polynomials**.

Prover

Prove it!

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample m random degree-$(d \cdot \ell)$ polynomials $\boldsymbol{P}_0(X) = \left(P_{0,1}(X), \ldots, P_{0,m}(X)\right)$

② Commit to the polynomials $\boldsymbol{P}_0, P_1, \ldots, P_n$

$\text{PCom}(\boldsymbol{P}_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomials $Q_1(X), \ldots, Q_m(X)$ such that

$$X \cdot Q_1(X) = X \cdot P_{0,1}(X) + f_1(P_1(X), \ldots, P_n(X))$$
$$\vdots$$
$$X \cdot Q_m(X) = X \cdot P_{0,m}(X) + f_m(P_1(X), \ldots, P_n(X))$$

$Q_1, \ldots, Q_m$ →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

← $\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

⑥ Check that $\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q_1(r_j) = r_j \cdot v_{j,0,1} + f_1(v_{j,1}, \ldots, v_{j,n})$$
$$\vdots$$
$$r_j \cdot Q_m(r_j) = r_j \cdot v_{j,0,m} + f_m(v_{j,1}, \ldots, v_{j,n}))$$

<u>Prover</u>

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample m random degree-$(d \cdot \ell)$ polynomials $\boldsymbol{P}_0(X) = \left(P_{0,1}(X), \ldots, P_{0,m}(X)\right)$

② Commit to the polynomials $\boldsymbol{P}_0, P_1, \ldots, P_n$

$\mathrm{PCom}(\boldsymbol{P}_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomials $Q_1(X), \ldots, Q_m(X)$ such that

$$X \cdot Q_1(X) = X \cdot P_{0,1}(X) + f_1(P_1(X), \ldots, P_n(X))$$
$$\vdots$$
$$X \cdot Q_m(X) = X \cdot P_{0,m}(X) + f_m(P_1(X), \ldots, P_n(X))$$

$Q_1, \ldots, Q_m$ →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q_1(r_j) = r_j \cdot v_{j,0,1} + f_1(v_{j,1}, \ldots, v_{j,n})$$
$$\vdots$$
$$r_j \cdot Q_m(r_j) = r_j \cdot v_{j,0,m} + f_m(v_{j,1}, \ldots, v_{j,n}))$$

Sigma/3-round variant of MQOM v2

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample m random degree-$(d \cdot \ell)$ polynomials $\boldsymbol{P}_0(X) = \left(P_{0,1}(X), \ldots, P_{0,m}(X)\right)$

② Commit to the polynomials $\boldsymbol{P}_0, P_1, \ldots, P_n$

$\mathrm{PCom}(\boldsymbol{P}_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomials $Q_1(X), \ldots, Q_m(X)$ such that

$$X \cdot Q_1(X) = X \cdot P_{0,1}(X) + f_1(P_1(X), \ldots, P_n(X))$$
$$\vdots$$
$$X \cdot Q_m(X) = X \cdot P_{0,m}(X) + f_m(P_1(X), \ldots, P_n(X))$$

$Q_1, \ldots, Q_m$ →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$r_1, \ldots, r_\ell$ ←

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q_1(r_j) = r_j \cdot v_{j,0,1} + f_1(v_{j,1}, \ldots, v_{j,n})$$
$$\vdots$$
$$r_j \cdot Q_m(r_j) = r_j \cdot v_{j,0,m} + f_m(v_{j,1}, \ldots, v_{j,n}))$$

A bit costly!

Prover

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample m random degree-$(d \cdot \ell)$ polynomials $\boldsymbol{P}_0(X) = \left(P_{0,1}(X), \ldots, P_{0,m}(X)\right)$

② Commit to the polynomials $\boldsymbol{P}_0, P_1, \ldots, P_n$

$\mathrm{PCom}(\boldsymbol{P}_0, P_1, \ldots, P_n)$ →

③ Reveal the polynomials $Q_1(X), \ldots, Q_m(X)$ such that

$$X \cdot Q_1(X) = X \cdot P_{0,1}(X) + f_1(P_1(X), \ldots, P_n(X))$$
$$\vdots$$
$$X \cdot Q_m(X) = X \cdot P_{0,m}(X) + f_m(P_1(X), \ldots, P_n(X))$$

$Q_1, \ldots, Q_m$ →

④ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$r_1, \ldots, r_\ell$ ←

⑤ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑥ Check that $\{\boldsymbol{v}_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

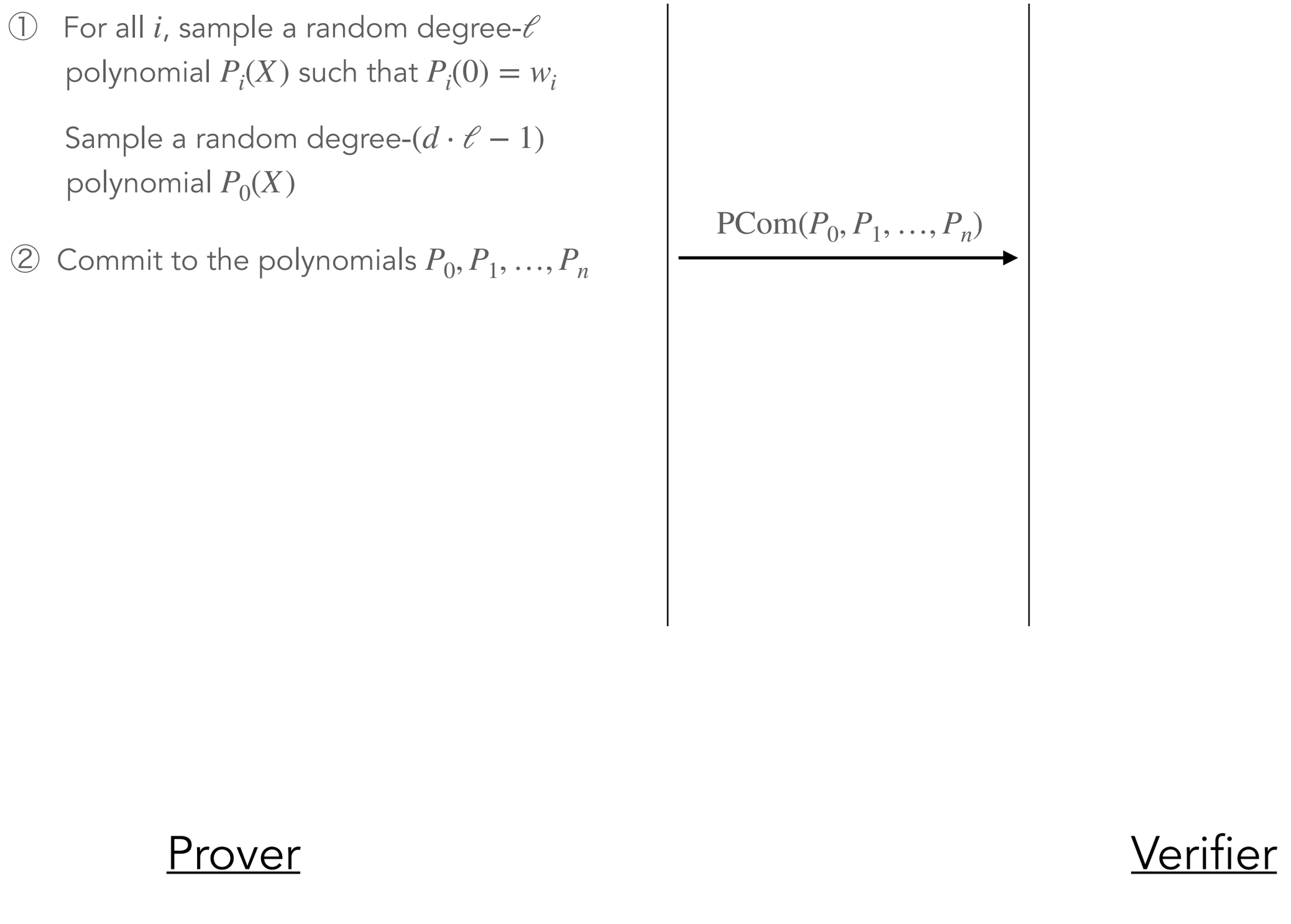
$$r_j \cdot Q_1(r_j) = r_j \cdot v_{j,0,1} + f_1(v_{j,1}, \ldots, v_{j,n})$$
$$\vdots$$
$$r_j \cdot Q_m(r_j) = r_j \cdot v_{j,0,m} + f_m(v_{j,1}, \ldots, v_{j,n}))$$

A bit costly!

Prover

Solution: batching

TCitH and VOLEitH Frameworks, *in the PIOP formalism*



TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

← $\gamma_1, \ldots, \gamma_m$

Q →

③ Choose random coefficients

$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Prover

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Choose random coefficients $\gamma_1, \ldots, \gamma_m \xleftarrow{\$} \mathbb{F}$ ←

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

Q →

Well-defined!

$$\sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(0), \ldots, P_n(0)) = \sum_{k=1}^{m} \gamma_k \cdot f_k(w_1, \ldots, w_n)$$

$$= \sum_{k=1}^{m} \gamma_k \cdot 0 = 0$$

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

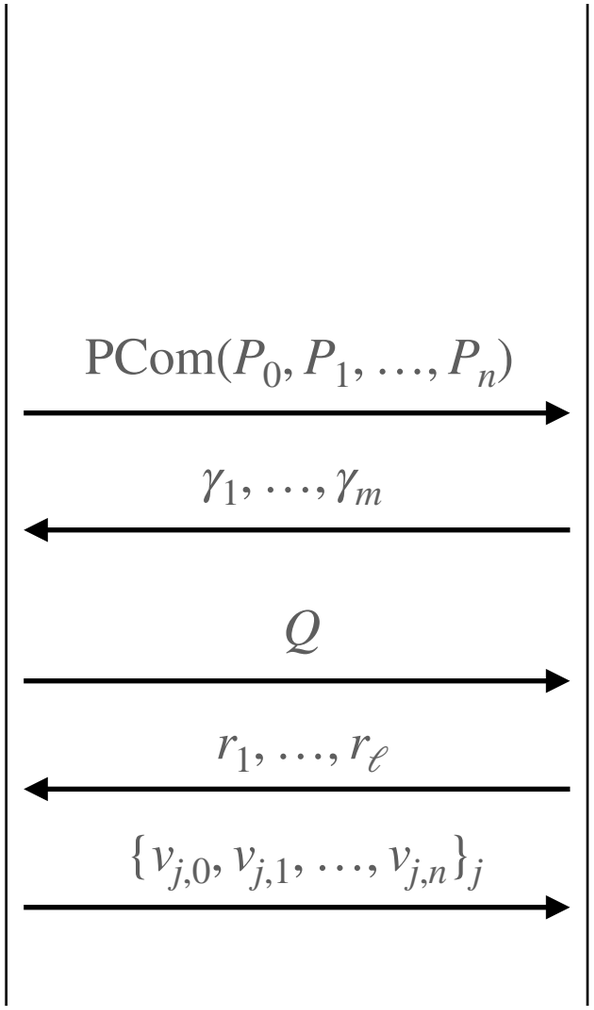
② Commit to the polynomials $P_0, P_1, \ldots, P_n$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$



③ Choose random coefficients

$$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$$

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Choose random coefficients

$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$

← $\gamma_1, \ldots, \gamma_m$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

→ Q

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

→ $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + \sum_{k=1}^{m} \gamma_k \cdot f_k(v_{j,1}, \ldots, v_{j,n})$$

Prover

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

Soundness Analysis

① For all i, choose a degree-ℓ polynomial $P_i(X)$. There exists j^* s.t.

$$f_{j^*}(P_1(0), \ldots, P_n(0)) \neq 0.$$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Choose random coefficients

$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$

← $\gamma_1, \ldots, \gamma_m$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

Q →

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + \sum_{k=1}^{m} \gamma_k \cdot f_k(v_{j,1}, \ldots, v_{j,n})$$

<u>Prover</u> <u>Verifier</u>

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, choose a degree-ℓ polynomial $P_i(X)$. There exists j^* s.t.

$$f_{j^*}(P_1(0), \ldots, P_n(0)) \neq 0.$$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\xrightarrow{\text{PCom}(P_0, P_1, \ldots, P_n)}$

$\xleftarrow{\gamma_1, \ldots, \gamma_m}$

③ Choose random coefficients $\gamma_1, \ldots, \gamma_m \xleftarrow{\$} \mathbb{F}$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

$\xrightarrow{Q}$

$\xleftarrow{r_1, \ldots, r_\ell}$

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$\xrightarrow{\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j}$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the

$v_{j,n})$

Soundness Analysis

It is an inequality with **high probability** over the randomness of $\gamma_1, \ldots, \gamma_m$, since we have

$$\sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(0), \ldots, P_n(0)) \neq 0$$

Prover

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, choose a degree-ℓ polynomial $P_i(X)$. There exists j^* s.t.

$$f_{j^*}(P_1(0), \ldots, P_n(0)) \neq 0.$$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\xrightarrow{\text{PCom}(P_0, P_1, \ldots, P_n)}$

$\xleftarrow{\gamma_1, \ldots, \gamma_m}$

③ Choose random coefficients $\gamma_1, \ldots, \gamma_m \xleftarrow{\$} \mathbb{F}$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

$\xrightarrow{Q}$

$\xleftarrow{r_1, \ldots, r_\ell}$

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

$\xrightarrow{\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j}$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Soundness Analysis

Schwartz-Zippel Lemma: Since it is a degree-$(d \cdot \ell)$ relation,

$$\Pr[\text{verification passes}] \leq \frac{\binom{d \cdot \ell}{\ell}}{\binom{|S|}{\ell}}.$$

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + \sum_{k=1}^{m} \gamma_k \cdot f_k(v_{j,1}, \ldots, v_{j,n})$$

Verifier

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

$\mathrm{PCom}(P_0, P_1, \ldots, P_n)$ →

← $\gamma_1, \ldots, \gamma_m$

Q →

← $r_1, \ldots, r_\ell$

$\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ →

③ Choose random coefficients

$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + \sum_{k=1}^{m} \gamma_k \cdot f_k(v_{j,1}, \ldots, v_{j,n})$$

Verifier

5-round variant used in most of the recent MPCitH-based signature schemes

TCitH and VOLEitH Frameworks, *in the PIOP formalism*

① For all i, sample a random degree-ℓ polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-$(d \cdot \ell - 1)$ polynomial $P_0(X)$

② Commit to the polynomials $P_0, P_1, \ldots, P_n$

$\text{PCom}(P_0, P_1, \ldots, P_n)$ →

③ Choose random coefficients

$\gamma_1, \ldots, \gamma_m \leftarrow^{\$} \mathbb{F}$

← $\gamma_1, \ldots, \gamma_m$

④ Reveal the polynomial $Q(X)$ such that

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{k=1}^{m} \gamma_k \cdot f_k(P_1(X), \ldots, P_n(X))$$

→ Q

⑤ Choose ℓ random evaluation points $r_1, \ldots, r_\ell \in \mathscr{C} \subset \mathbb{F}$

← $r_1, \ldots, r_\ell$

⑥ For all (i, j), reveal the evaluation

$$v_{j,i} := P_i(r_j)$$

→ $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$

⑦ Check that $\{v_{j,0}, v_{j,1}, \ldots, v_{j,n}\}_j$ are consistent with the commitment.

Check that, for all j,

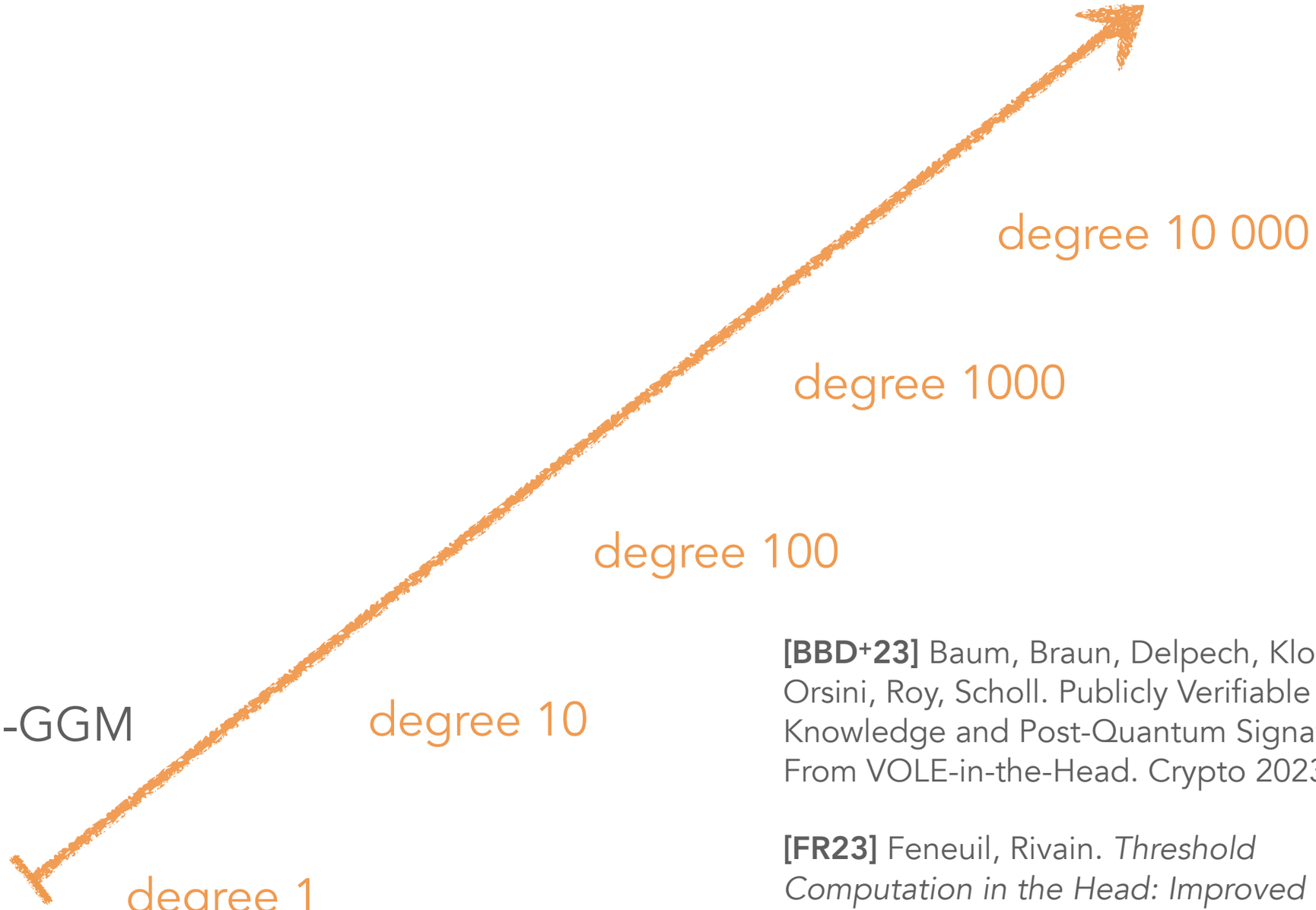
$$r_j \cdot Q(r_j) = r_j \cdot v_{j,0} + \sum_{k=1}^{m} \gamma_k \cdot f_k(v_{j,1}, \ldots, v_{j,n})$$

<u>Prover</u> <u>Verifier</u>

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.



① VOLEitH / TCitH-GGM

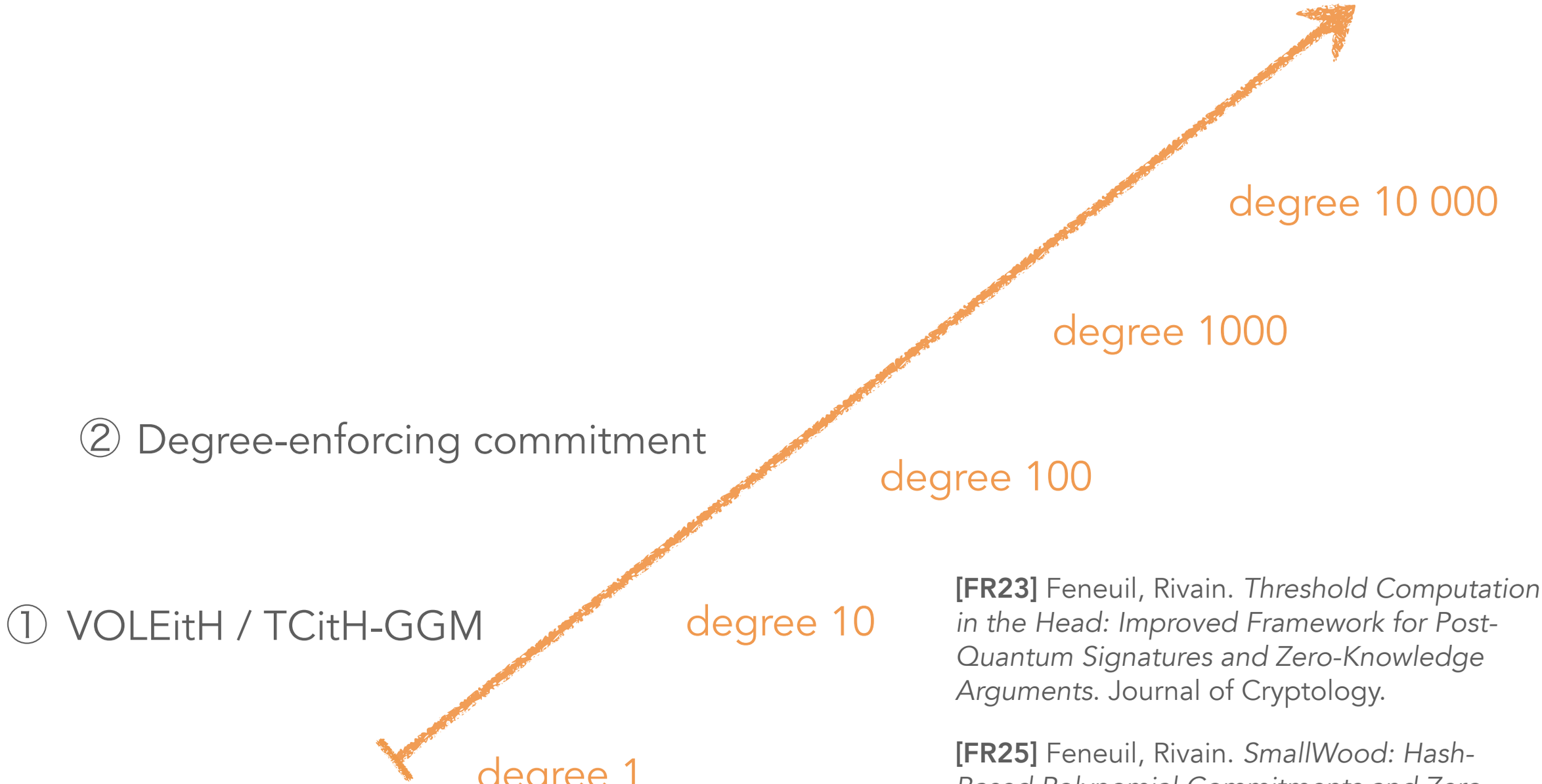
[BBD+23] Baum, Braun, Delpech, Klooß, Orsini, Roy, Scholl. Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head. Crypto 2023.

[FR23] Feneuil, Rivain. *Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments*. Journal of Cryptology.

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.



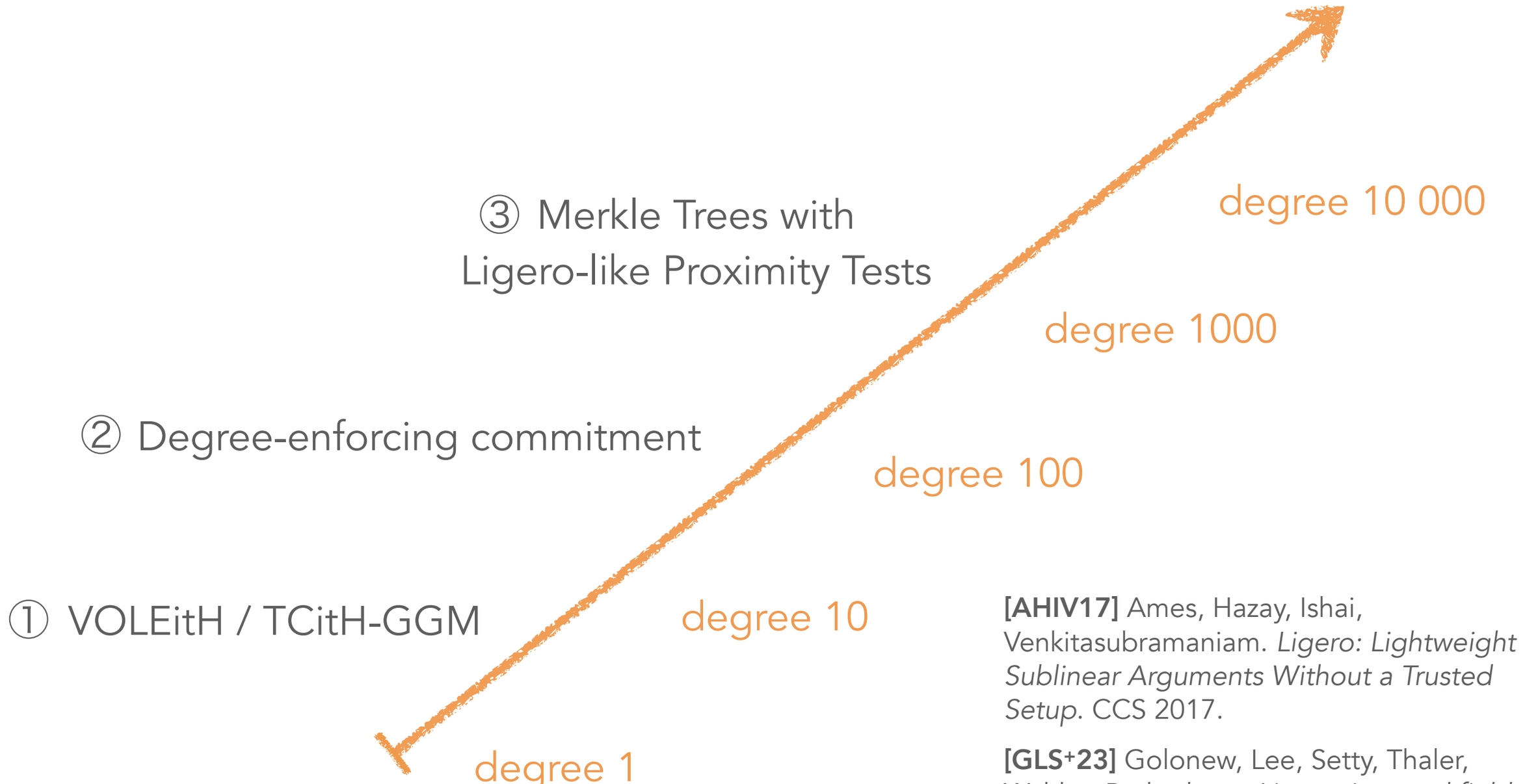
[FR23] Feneuil, Rivain. *Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments*. Journal of Cryptology.

[FR25] Feneuil, Rivain. *SmallWood: Hash-Based Polynomial Commitments and Zero-Knowledge Arguments for Relatively Small Instances*. ePrint 2025/1085.

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.



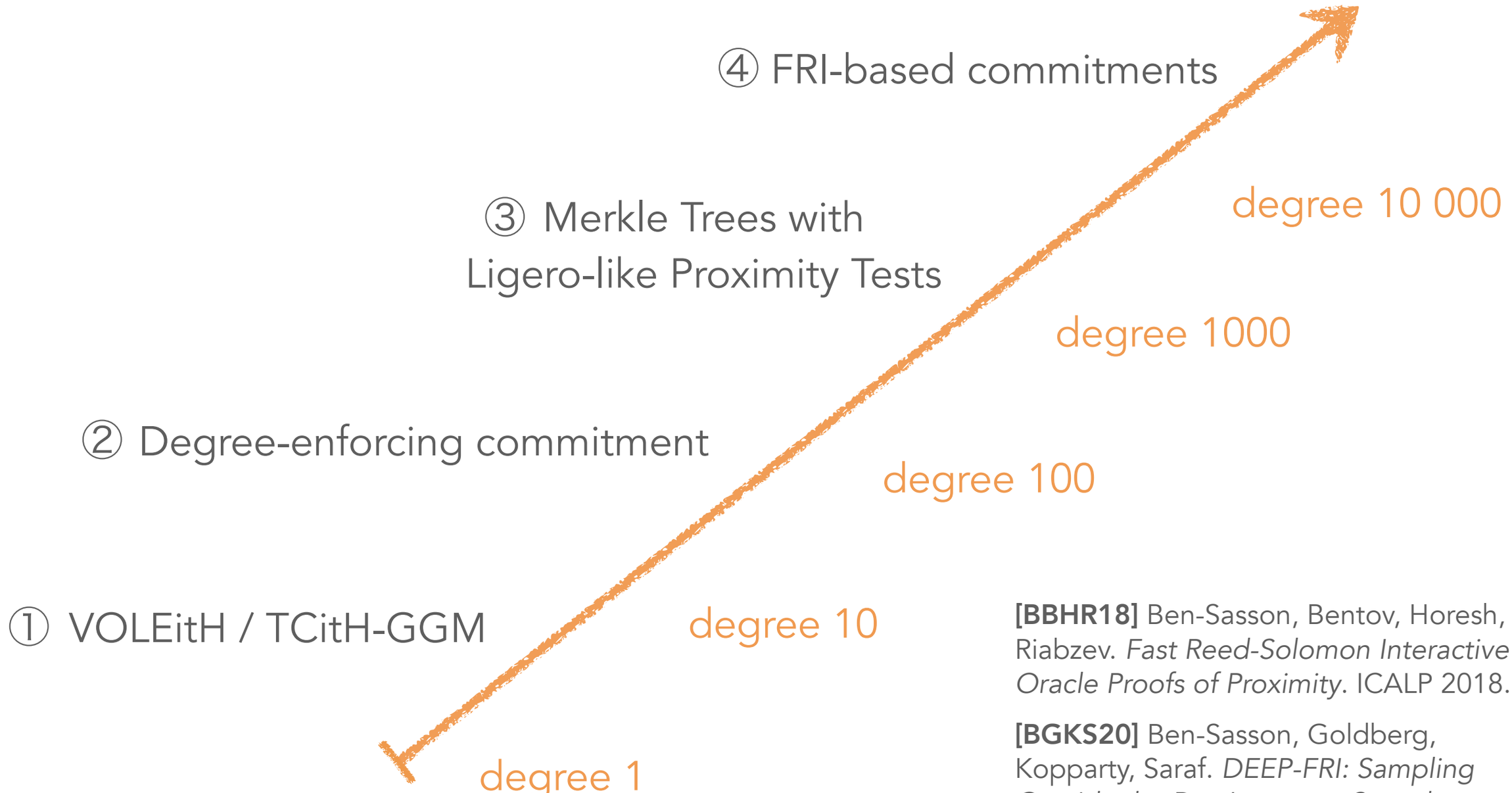
[AHIV17] Ames, Hazay, Ishai, Venkitasubramaniam. *Ligero: Lightweight Sublinear Arguments Without a Trusted Setup*. CCS 2017.

[GLS+23] Golonew, Lee, Setty, Thaler, Wahby. *Brakedown: Linear-time and field agnotic SNARKs for R1CS*. Crypto 2023.

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.



[BBHR18] Ben-Sasson, Bentov, Horesh, Riabzev. *Fast Reed-Solomon Interactive Oracle Proofs of Proximity*. ICALP 2018.

[BGKS20] Ben-Sasson, Goldberg, Kopparty, Saraf. *DEEP-FRI: Sampling Outside the Box Improves Soundness*. ITCS 2020.

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.

Merkle Tree

④ FRI-based commitments

③ Merkle Trees with
Ligero-like Proximity Tests

② Degree-enforcing commitment

① VOLEitH / TCitH-GGM

GGM Tree

degree 10 000

degree 1000

degree 100

degree 10

degree 1

How to commit to polynomials?

(using symmetric primitives)

Values are indicative only and serve to illustrate the progression of the scale.

Merkle Tree

④ FRI-based commitments

③ Merkle Trees with
Ligero-like Proximity Tests

② Degree-enforcing commitment

① VOLEitH / TCitH-GGM

GGM Tree

degree 1

degree 10

degree 100

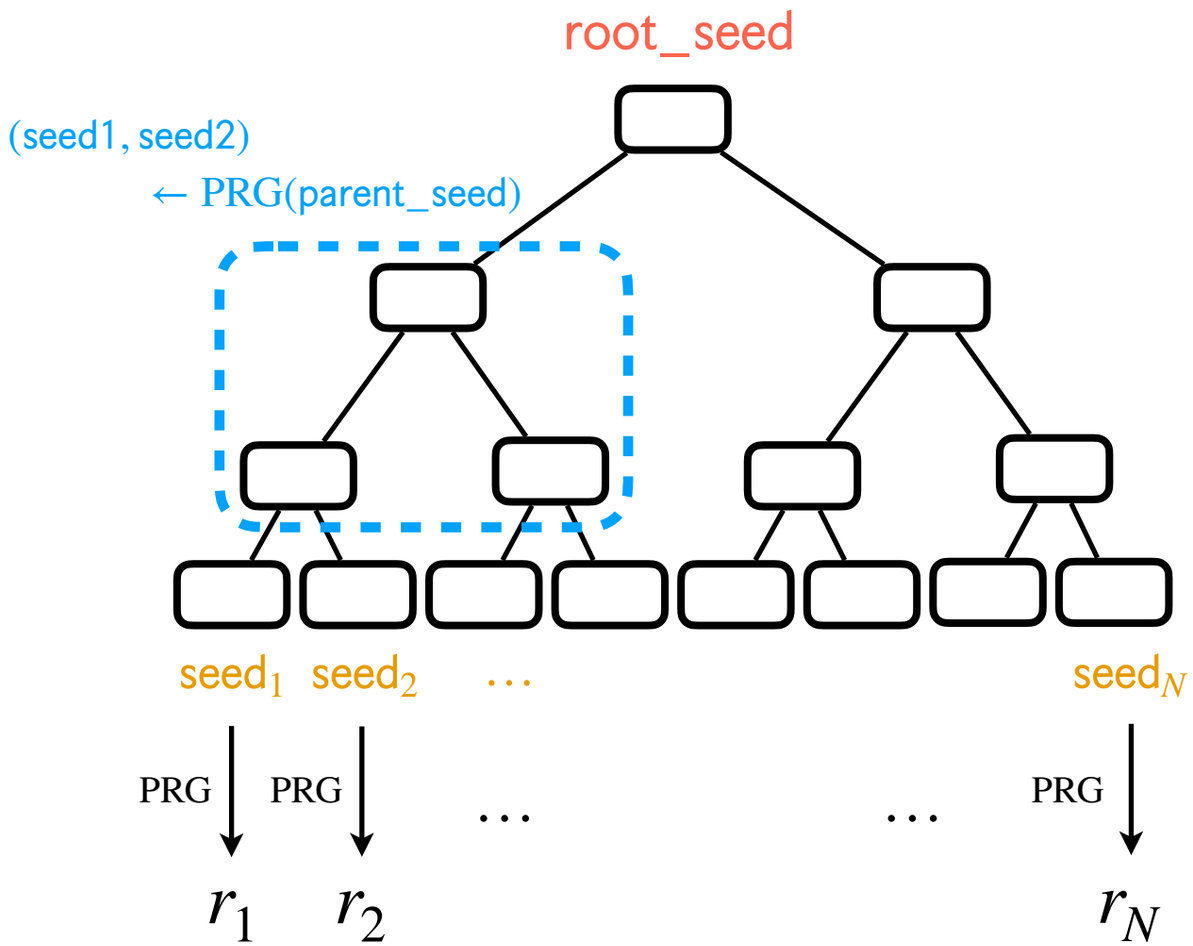
degree 1000

degree 10 000

For signature schemes, we use
degree-1 polynomials most of the time.

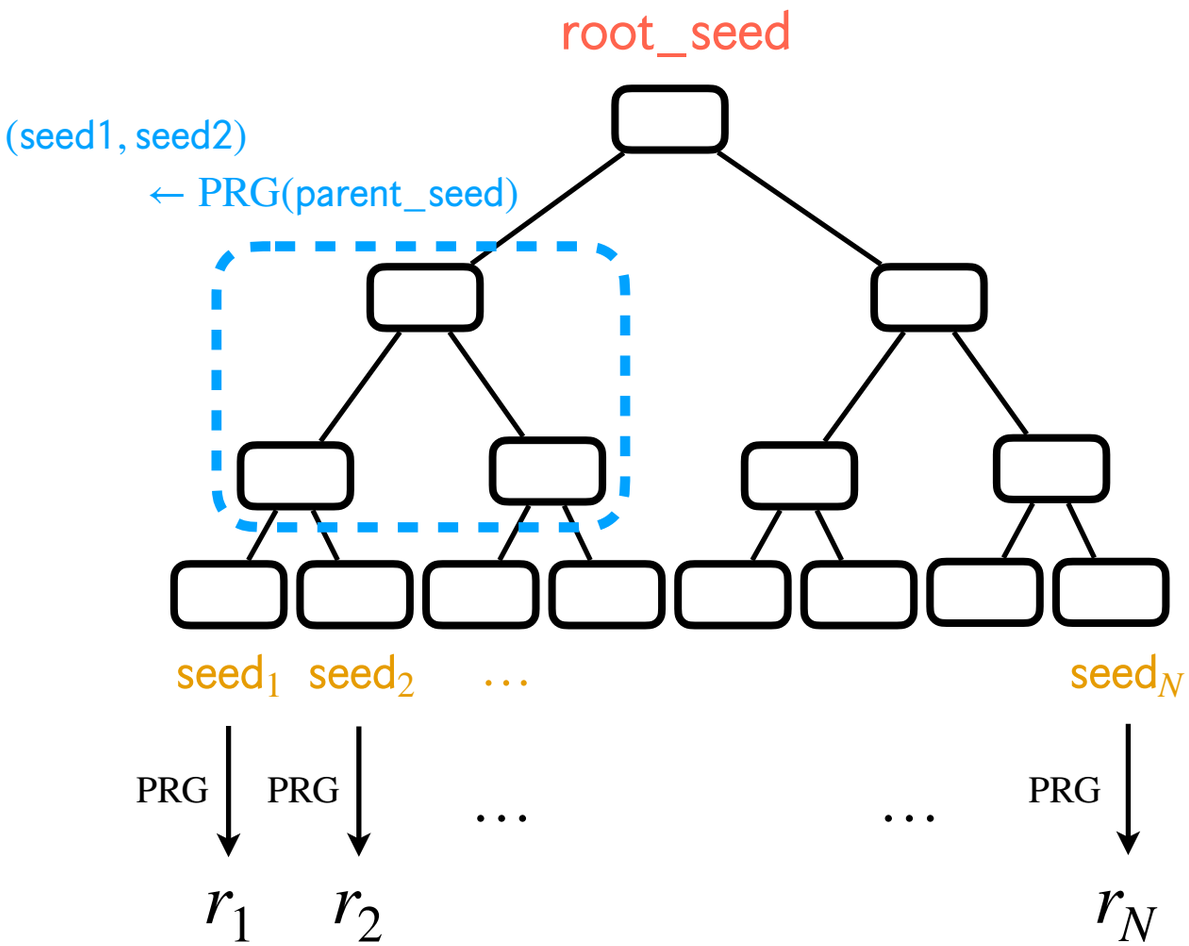
Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



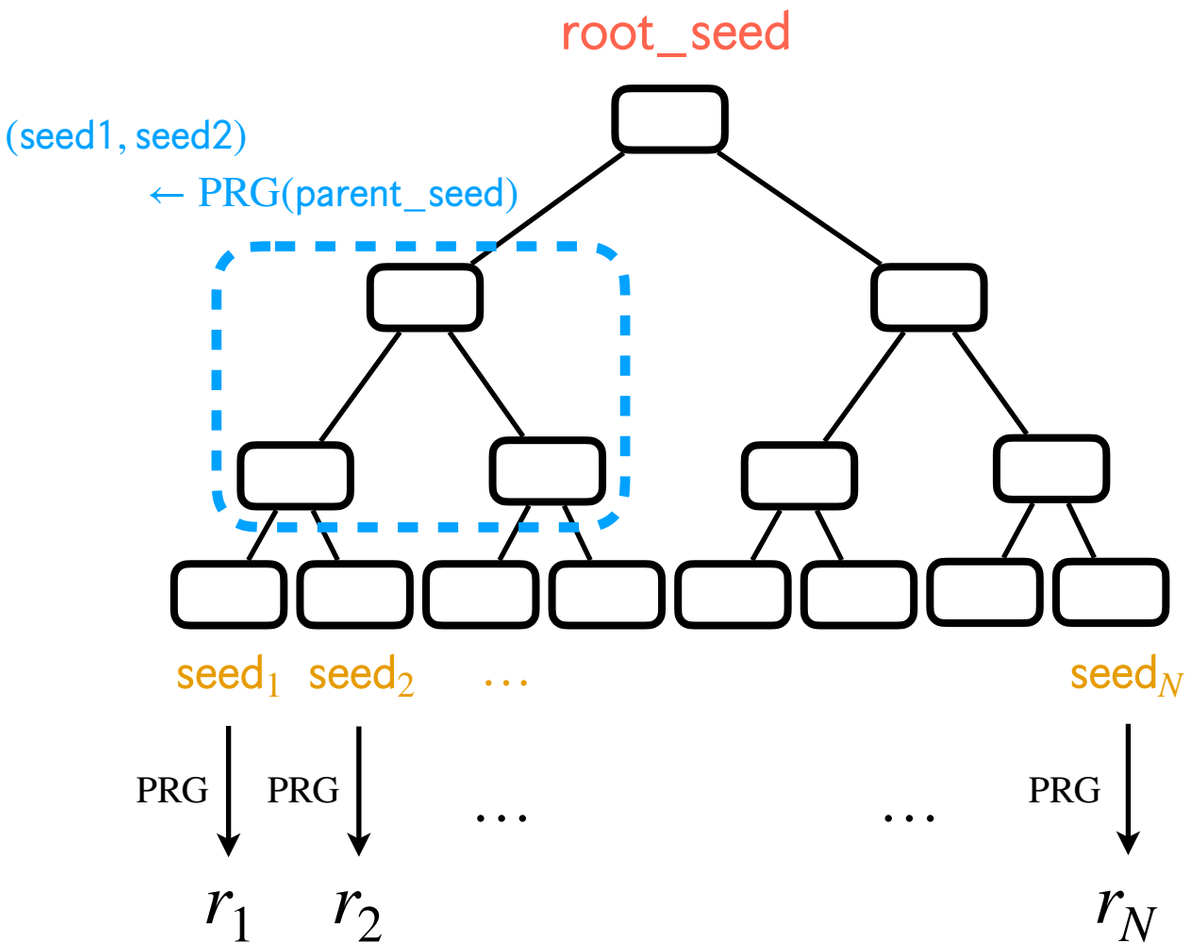
Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \sum_{i=1}^{N} r_i \cdot (X - e_i)$$

(assuming $\deg P = 1$*)*

Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



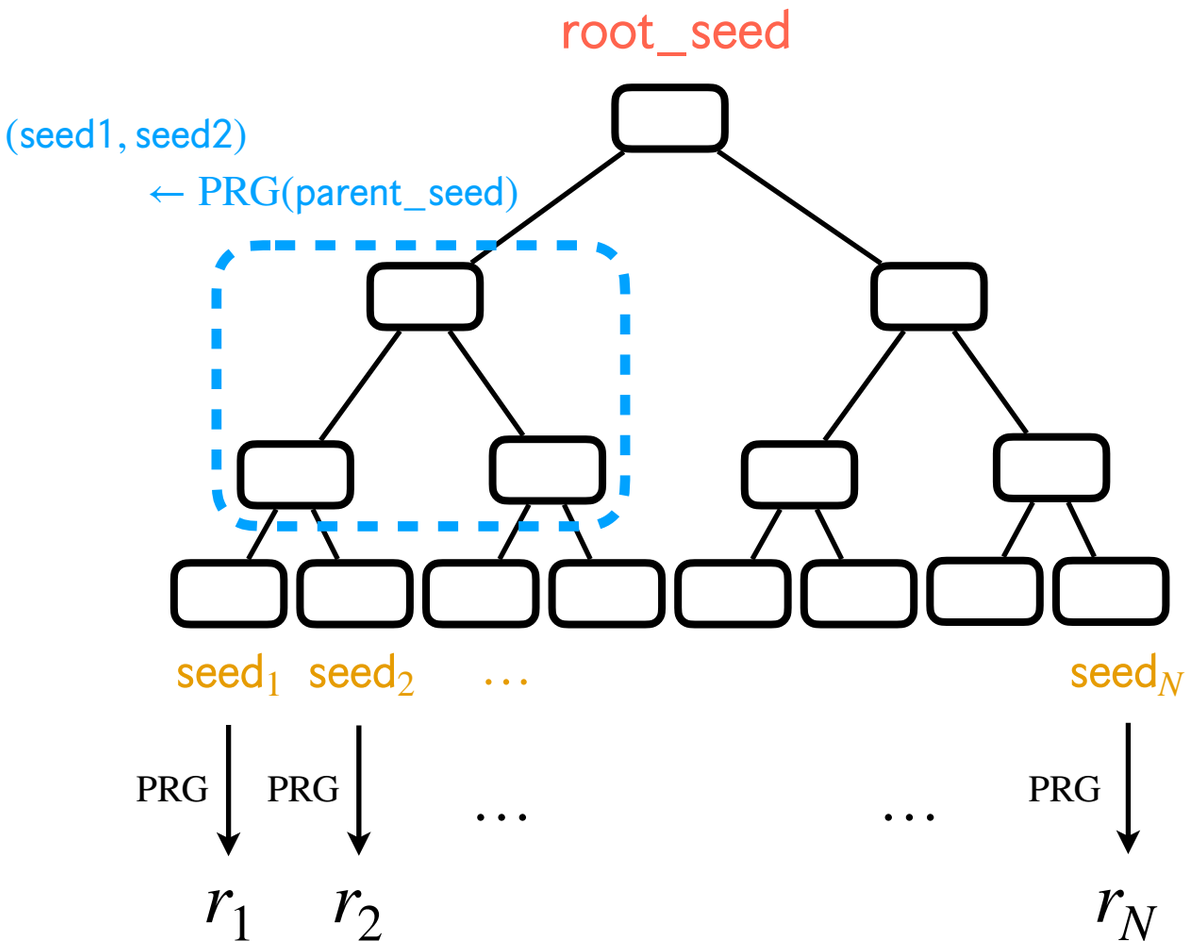
Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \underbrace{\sum_{i=1}^{N} r_i \cdot (X - e_i)}_{\text{Mask}}$$

(assuming $\deg P = 1$*)*

Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \underbrace{\sum_{i=1}^{N} r_i \cdot (X - e_i)}_{\text{Mask}}$$

(assuming $\deg P = 1$*)*

Commitment:

- *Commit to each seed* ***independently***
- *Reveal the masked polynomial* $\Delta P(X)$

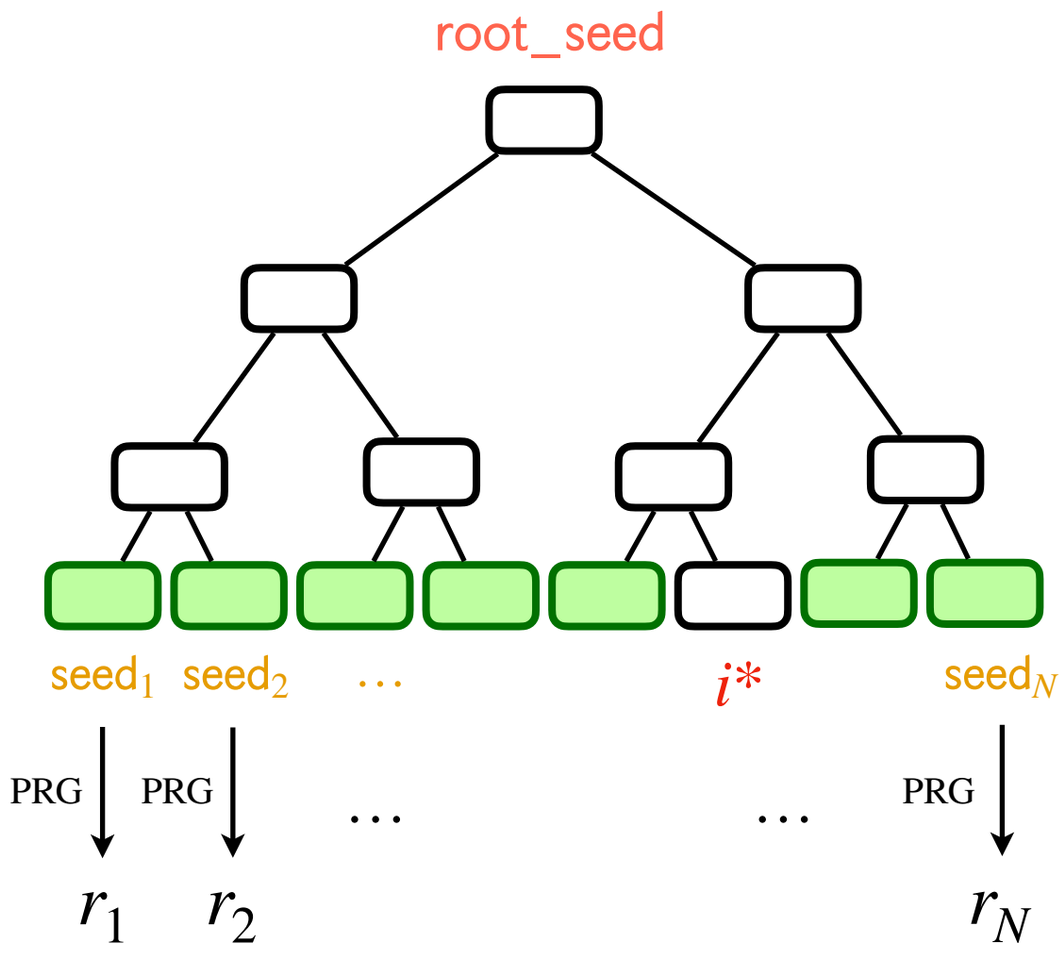
Open $P(e_{i^*})$*:*

Reveal all $\{r_i\}_{i \neq i^*}$ *since*

$$P(e_{i^*}) = \Delta P(e_{i^*}) + \sum_{i \neq i^*} r_i \cdot (e_{i^*} - e_i)$$

Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



Commitment:

- *Commit to each seed* ***independently***
- *Reveal the masked polynomial* $\Delta P(X)$

Open $P(e_{i^*})$:

Reveal all $\{r_i\}_{i \neq i^*}$ *since*

$$P(e_{i^*}) = \Delta P(e_{i^*}) + \sum_{i \neq i^*} r_i \cdot (e_{i^*} - e_i)$$

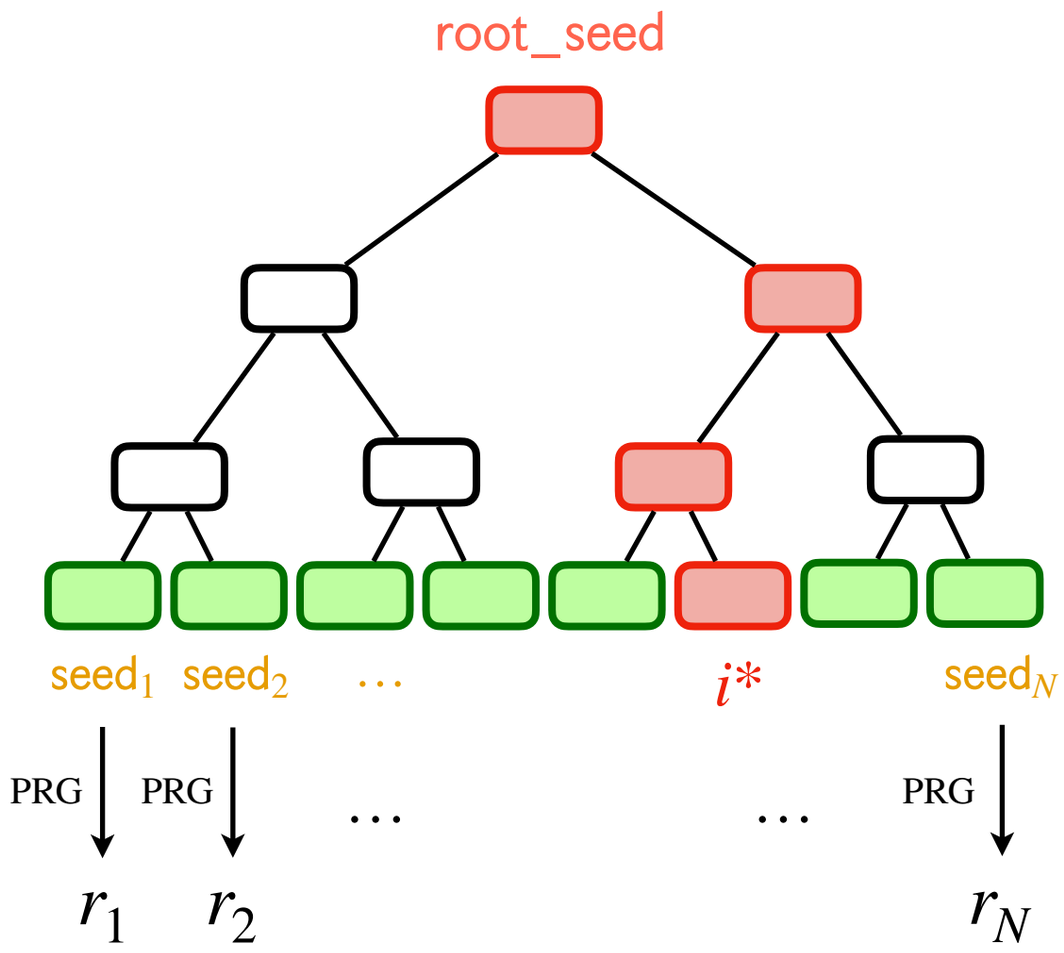
Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \underbrace{\sum_{i=1}^{N} r_i \cdot (X - e_i)}_{\text{Mask}}$$

(assuming $\deg P = 1$*)*

Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \underbrace{\sum_{i=1}^{N} r_i \cdot (X - e_i)}_{\text{Mask}}$$

(assuming $\deg P = 1$*)*

Commitment:

- *Commit to each seed* ***independently***
- *Reveal the masked polynomial* $\Delta P(X)$

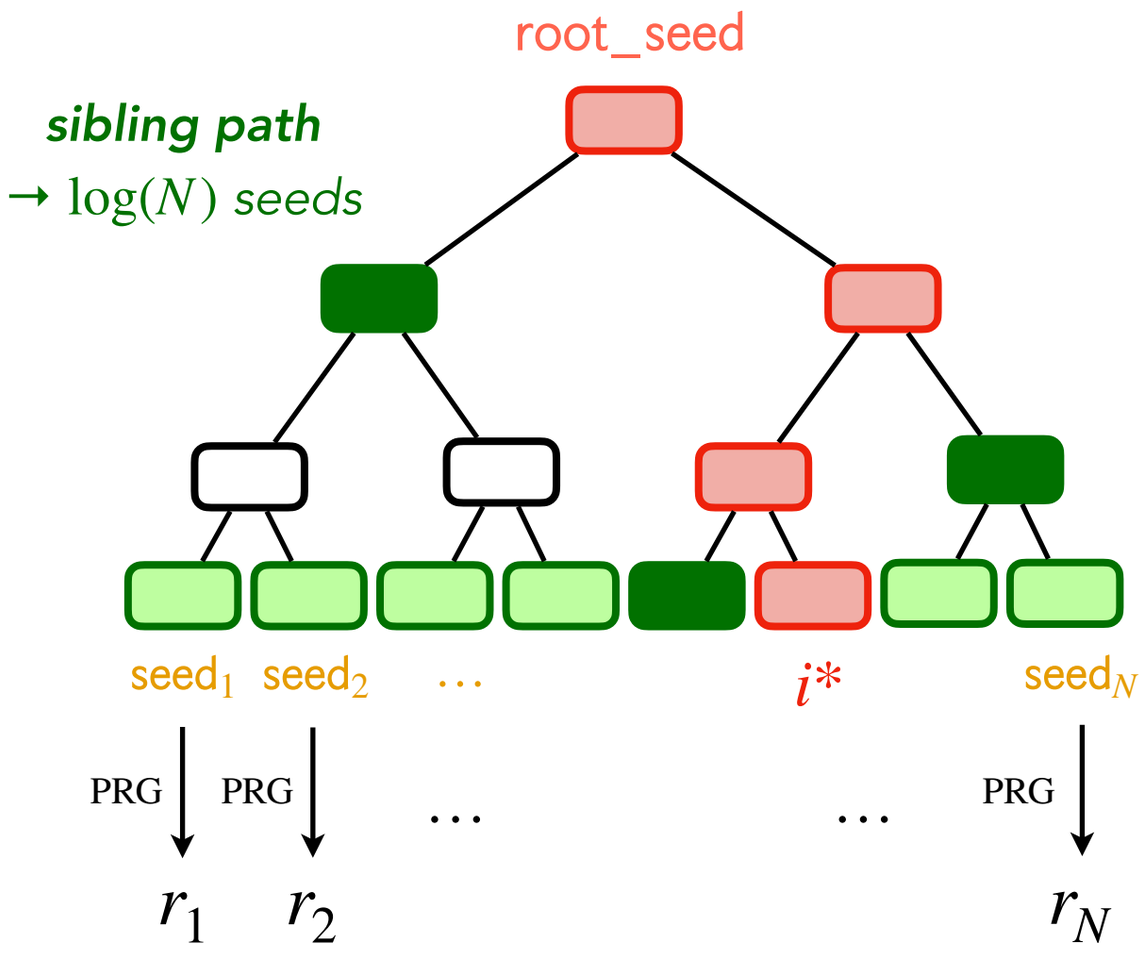
Open $P(e_{i^*})$*:*

Reveal all $\{r_i\}_{i \neq i^*}$ *since*

$$P(e_{i^*}) = \Delta P(e_{i^*}) + \sum_{i \neq i^*} r_i \cdot (e_{i^*} - e_i)$$

Using a GGM tree (*ie.* a seed tree)

[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



Build $\Delta P(X)$ as

$$\Delta P(X) := P(X) - \underbrace{\sum_{i=1}^{N} r_i \cdot (X - e_i)}_{\text{Mask}}$$

(assuming $\deg P = 1$*)*

Commitment:

- *Commit to each seed* ***independently***
- *Reveal the masked polynomial* $\Delta P(X)$

Open $P(e_{i*})$*:*

Reveal all $\{r_i\}_{i \neq i*}$ *since*

$$P(e_{i*}) = \Delta P(e_{i*}) + \sum_{i \neq i*} r_i \cdot (e_{i*} - e_i)$$

Committing to a Polynomial using a Seed Tree

A seed tree of N leaves to commit to degree-1 polynomials

- 👉 The prover can provably open N evaluations (i.e. $N = |\mathscr{C}|$)
- 👉 Soundness error of $\frac{d}{N}$

$$\frac{\binom{d \cdot \ell}{\ell}}{\binom{|\mathscr{C}|}{\ell}} \text{ with } \ell := 1$$

Committing to a Polynomial using a Seed Tree

A seed tree of N leaves to commit to degree-1 polynomials

- 👉 The prover can provably open N evaluations (i.e. $N = |\mathscr{C}|$)
- 👉 Soundness error of $\frac{d}{N}$

How to have a negligible soundness error? 🤔

Committing to a Polynomial using a Seed Tree

A seed tree of N leaves to commit to degree-1 polynomials

- 👉 The prover can provably open N evaluations (i.e. $N = |\mathscr{C}|$)
- 👉 Soundness error of $\frac{d}{N}$

How to have a negligible soundness error? 🤔

1. Taking $N \geq 2^{\lambda}$. Impossible since the complexity would be in $O(2^{\lambda})$.

Committing to a Polynomial using a Seed Tree

A seed tree of N leaves to commit to degree-1 polynomials

- 👉 The prover can provably open N evaluations (i.e. $N = |\mathscr{C}|$)
- 👉 Soundness error of $\frac{d}{N}$

How to have a negligible soundness error? 🤔

1. Taking $N \geq 2^{\lambda}$. Impossible since the complexity would be in $O(2^{\lambda})$.

2. *TCitH-GGM Approach*. Taking N small (e.g. $N = 256$) and repeating the protocol τ times. Soundness error of $\left(\frac{d}{N}\right)^{\tau}$.

Committing to a Polynomial using a Seed Tree

A seed tree of N leaves to commit to degree-1 polynomials

- 👉 The prover can provably open N evaluations (i.e. $N = |\mathscr{C}|$)
- 👉 Soundness error of $\frac{d}{N}$

How to have a negligible soundness error? 🤔

1. Taking $N \geq 2^{\lambda}$. Impossible since the complexity would be in $O(2^{\lambda})$.

2. *TCitH-GGM Approach*. Taking N small (e.g. $N = 256$) and repeating the protocol τ times. Soundness error of $\left(\frac{d}{N}\right)^{\tau}$.

3. *VOLEitH Approach*. Embed τ polynomials over $\mathbb{F}_q$ into a unique polynomial over $\mathbb{F}_{q^{\tau}}$, for which we will be able to open N^{τ} evaluations. Soundness error of $\frac{d}{N^{\tau}}$.

Arithmetization

How to build signature schemes?

Private key: solution of a degree-d multivariate system

How to build signature schemes?

Private key: solution of a degree-d multivariate system

PIOP
for polynomial constraints

Low-degree polynomials
Often degree-1 polynomials

How to build signature schemes?

Private key: solution of a degree-d multivariate system

PIOP
for polynomial constraints

Low-degree polynomials
Often degree-1 polynomials

PCS
from GGM trees

Soundness Error: $\frac{d}{N}$

with N the size of the tree

How to build signature schemes?

Private key: solution of a degree-d multivariate system

PIOP for polynomial constraints

Low-degree polynomials
Often degree-1 polynomials

PCS from GGM trees

Soundness Error: $\frac{d}{N}$

with N the size of the tree

Parallel repetitions

Soundness Error: $\left(\frac{d}{N}\right)^{\tau}$

Using VOLEitH technique

Soundness Error: $\frac{d}{N^{\tau}}$

How to build signature schemes?

Private key: solution of a degree-d multivariate system

PIOP for polynomial constraints

Low-degree polynomials
Often degree-1 polynomials

↓

PCS from GGM trees

Soundness Error: $\frac{d}{N}$

with N the size of the tree

↙ ↘

Parallel repetitions

Soundness Error: $\left(\frac{d}{N}\right)^{\tau}$

Using VOLEitH technique

Soundness Error: $\frac{d}{N^{\tau}}$

↘ ↙

Fiat-Shamir transformation

Arithmetization

Multivariate cryptography

Arithmetization

Multivariate cryptography

Public key: a multivariate quadratic (MQ) system $\mathcal{F}$

Private key: a solution of the MQ system

$x \in \mathbb{F}^n$ *such that* $\mathcal{F}(x) = 0$

Arithmetization

Multivariate cryptography

Public key: a multivariate quadratic (MQ) system $\mathscr{F}$

Private key: a solution of the MQ system

$$x \in \mathbb{F}^n \text{ such that } \mathscr{F}(x) = 0$$

👉 *We can directly apply the previous construction.*

Design of the MQOM scheme

Arithmetization

Rank-based cryptography

Arithmetization

Rank-based cryptography

Public key: a list of k matrices $M_1, \ldots, M_k \in \mathbb{F}^{m \times n}$

Private key: a vector $x \in \mathbb{F}^k$ such that

$$\sum_i x_i \cdot M_i \text{ is of rank at most } r$$

Not a polynomial constraint over the secret

Arithmetization

Rank-based cryptography

Public key: a list of k matrices $M_1, \ldots, M_k \in \mathbb{F}^{m \times n}$

Private key: a vector $x \in \mathbb{F}^k$ such that

$$\sum_i x_i \cdot M_i \text{ is of rank at most } r$$

Proved statement: there exists two matrices $L \in \mathbb{F}^{m \times r}$ and $R \in \mathbb{F}^{r \times n}$ such that

$$L \cdot R = \sum_i x_i \cdot M_i.$$

Degre-2 Polynomial Constraints

Design of the Mirath scheme

Arithmetization

Lattice-based cryptography

Public key: a matrix $A \in \mathbb{F}^{m \times n}$ *and a vector* $u \in \mathbb{F}^m$

Private key: a vector $x \in \mathbb{F}^k$ *such that*

$$u = Ax \qquad \text{and} \qquad \|x\|_\infty \leq \beta$$

Arithmetization

Lattice-based cryptography

Public key: a matrix $A \in \mathbb{F}^{m \times n}$ and a vector $u \in \mathbb{F}^m$

Private key: a vector $x \in \mathbb{F}^k$ such that

$$u = Ax \quad \text{and} \quad \|x\|_\infty \leq \beta$$

Proved statement: one can show that $u = Ax$ and for all i,

$$R(x_i) = 0$$

where $R := (X + \beta) \cdot (X + \beta - 1) \cdot \ldots \cdot (X - \beta)$.

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

linear, easy to check

non-linear, hard to check

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Proved statement: x satisfies $y = Hx$ and

there exists Q a degree-$(w-1)$ polynomial such that,
for all $i \in \{1,\ldots,n\}$,

$$x_i \cdot (\omega_i^w + Q(\omega_i)) = 0,$$

where $\omega_1, \ldots, \omega_n$ are distinct public values.

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Proved statement: x satisfies $y = Hx$ and

there exists Q a degree-$(w-1)$ polynomial such that,

for all $i \in \{1,\ldots,n\}$,

$$x_i \cdot (\omega_i^w + Q(\omega_i)) = 0,$$

Can be zero only for at most w values

where $\omega_1, \ldots, \omega_n$ are distinct public values.

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Proved statement: x satisfies $y = Hx$ and

there exists Q a degree-$(w-1)$ polynomial such that,
for all $i \in \{1,\ldots,n\}$,

$$x_i \cdot (\omega_i^w + Q(\omega_i)) = 0,$$

where $\omega_1, \ldots, \omega_n$ are distinct public values.

Design of the SDitH scheme (v1)

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Design of the SDitH scheme (v2)

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Proved statement: x satisfies $y = Hx$ and

$$x := e_1 + \ldots + e_w$$

where $e_1, \ldots, e_w$ are elementary vectors, which can thus be expressed as tensor products of smaller elementary vectors.

Design of the SDitH scheme (v2)

Arithmetization

Code-based cryptography

Public key: a matrix $H \in \mathbb{F}^{(n-k)\times n}$ and a vector $y \in \mathbb{F}^{n-k}$

Private key: a vector $x \in \mathbb{F}^n$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Example:

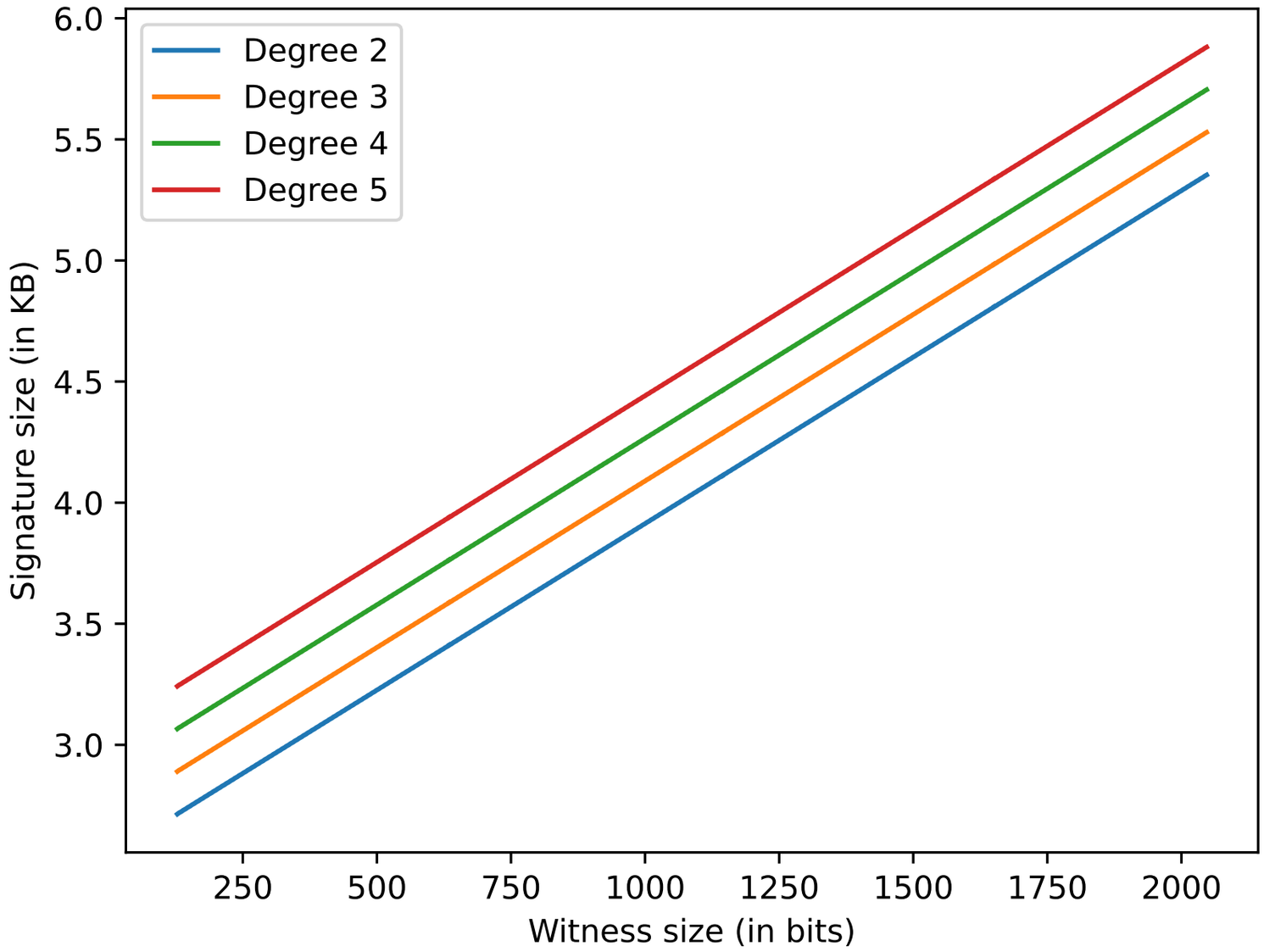
$$x = 01001000 = 01000000 + 00001000$$

$$\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \qquad \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix}$$

Design of the SDitH scheme (v2)

Arithmetization

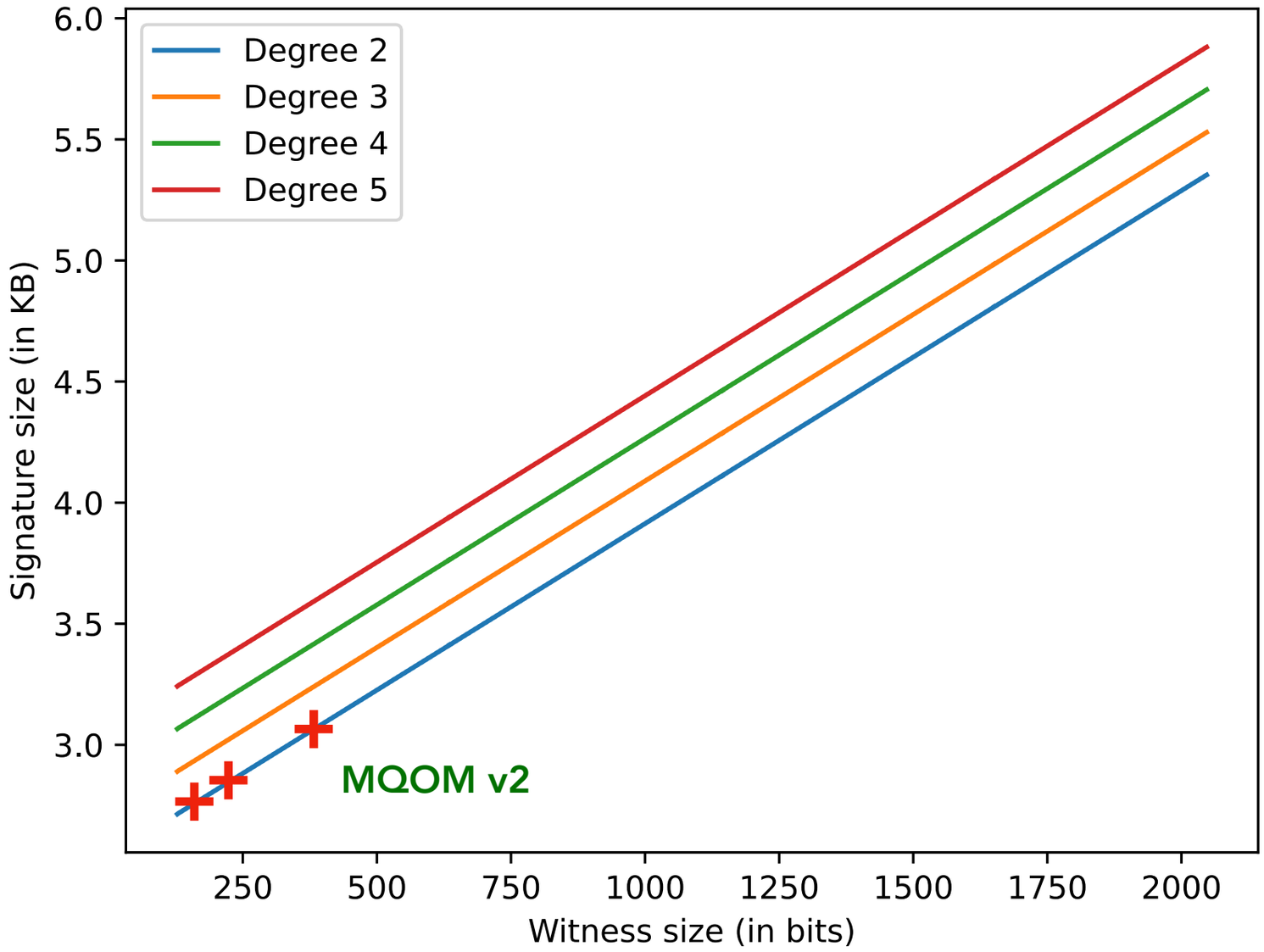
When using PIOP-based MPCitH frameworks, the goal is to find a modeling with the smallest witness and relatively small degree.



Using seed trees of around 2048 leaves

Arithmetization

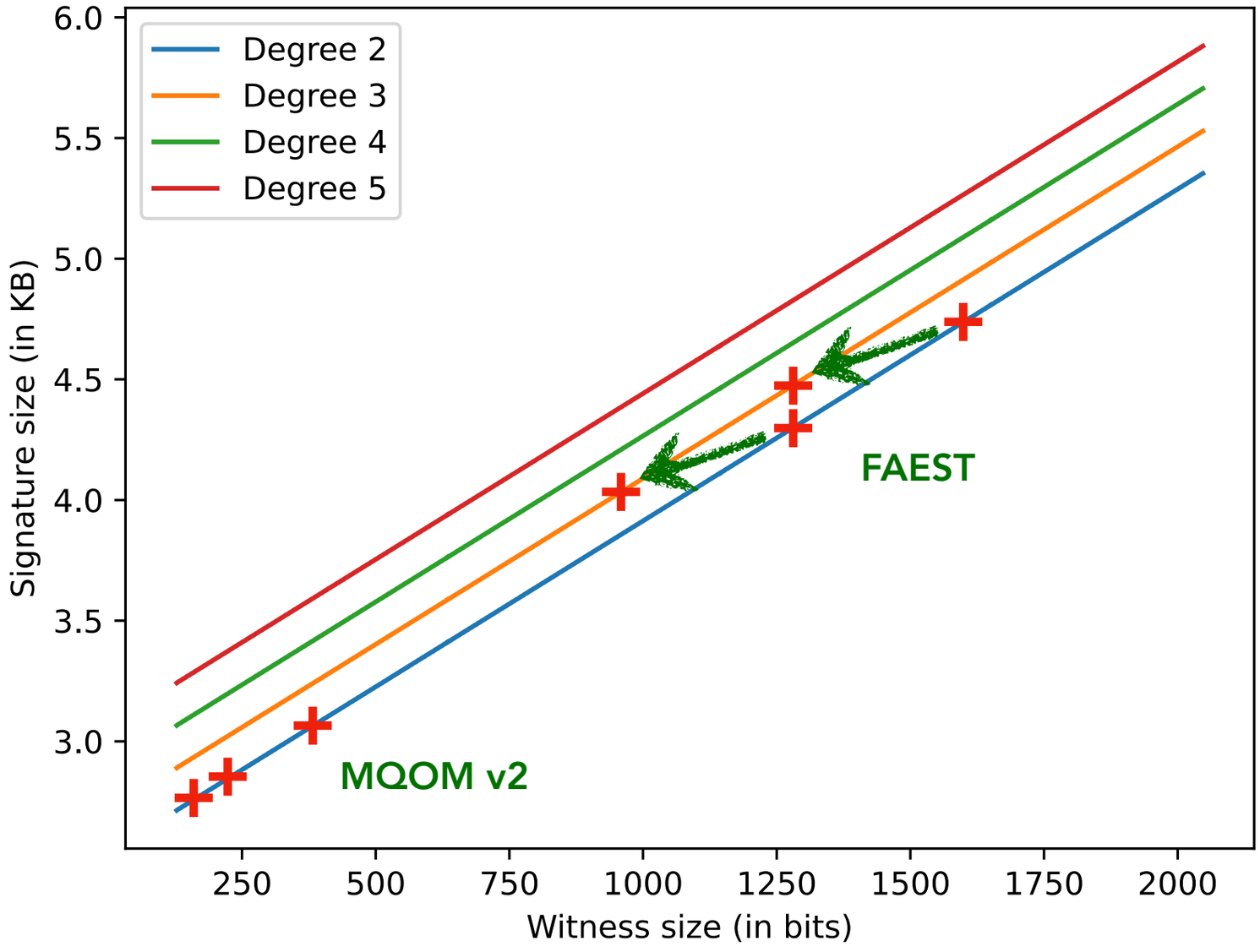
When using PIOP-based MPCitH frameworks, the goal is to find a modeling with the smallest witness and relatively small degree.



Using seed trees of around 2048 leaves

Arithmetization

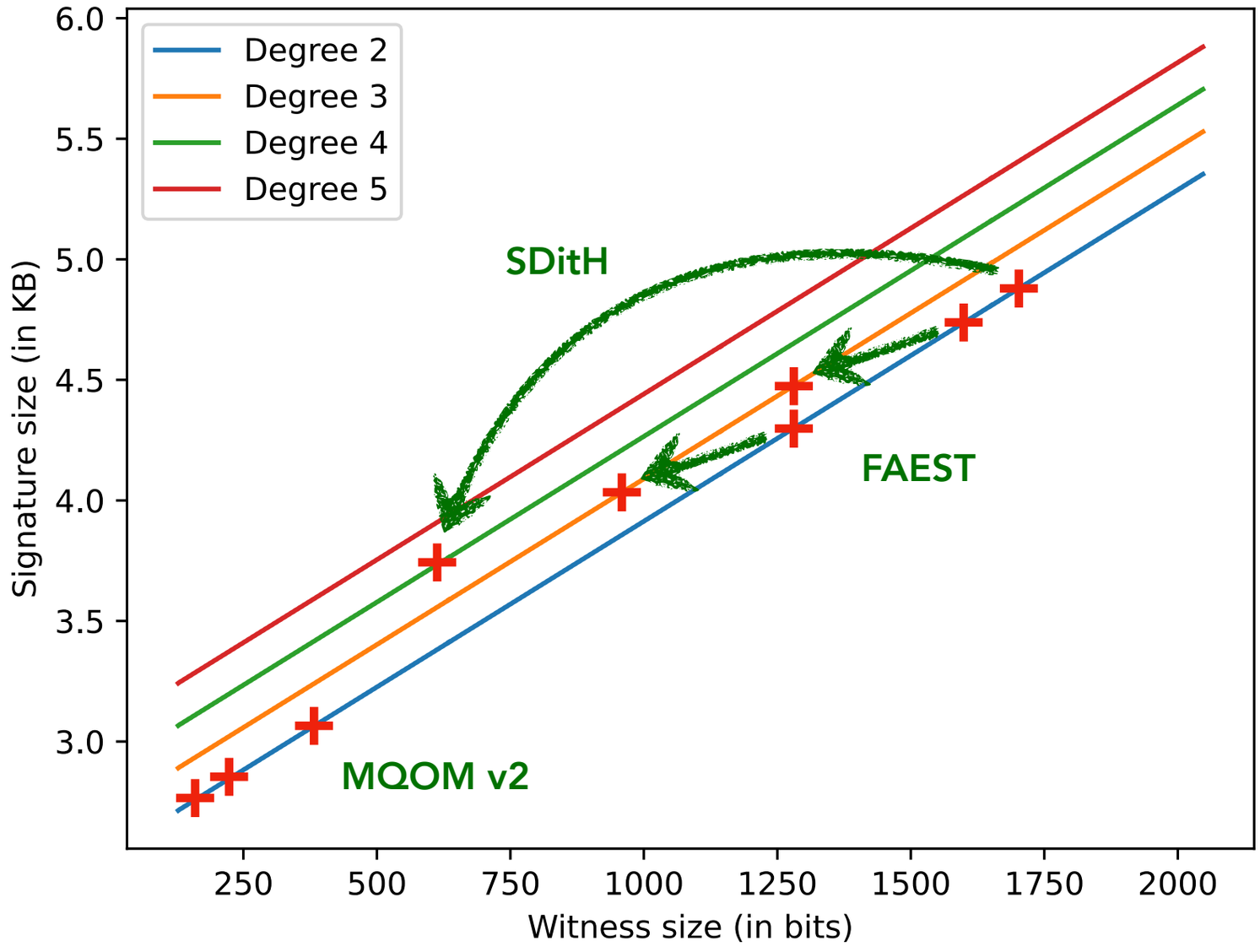
When using PIOP-based MPCitH frameworks, the goal is to find a modeling with the smallest witness and relatively small degree.



Using seed trees of around 2048 leaves

Arithmetization

When using PIOP-based MPCitH frameworks, the goal is to find a modeling with the smallest witness and relatively small degree.



Using seed trees of around 2048 leaves

For the sake of diversity…

AES Key Recovery
AIM Key Recovery
LowMC Key Recovery
Rain Key Recovery

Anemoi Hash Preimage
Poseidon Hash Preimage
Griffin Hash Preimage
RescuePrime Hash Preimage

LWE
SIS
Subset Sum

Syndrome Decoding
Regular Syndrome Decoding
Permuted Kernel
Linear Code Equivalence
Restricted Syndrome Decoding

MinRank
Rank Syndrome Decoding
Subfield Collision Problem
Matrix Subcode Equivalence

Multivariate Quadratic
PowAff2

Legendre PRF
BHHG's PRF

Discrete Logarithm
Integer Factorization
Double Discrete Logarithm

To use the PIOP-based MPCitH frameworks,
one just needs to write those problems using polynomial constraints.

NIST Candidates

	NIST Submission		
Security Assumptions	***Candidate Name***	***Sig. Size***	***PK Size***
AES Block cipher	FAEST v2	3.9-4.5 KB	32 B
Multivariate Quadratic	MQOM v2	2.8-3.2 KB	52-80 B
Syndrome Decoding	SDitH v2	3.7 KB	70 B
MinRank	Mirath	3.0-3.2 KB	57-73 B
Permuted Kernel	PERK v2.1	3.5 KB	100 B
Rank Syndrome Decoding	RYDE v2	3.1 KB	69 B

Round-3

Round-2

Using seed trees of around 2048 leaves

NIST Candidates

	NIST Submission		
Security Assumptions	***Candidate Name***	***Sig. Size***	***PK Size***
AES Block cipher	FAEST v2	3.9-4.5 KB	32 B
Multivariate Quadratic	MQOM v2	2.8-3.2 KB	52-80 B
Syndrome Decoding	SDitH v2	3.7 KB	70 B
MinRank	Mirath	3.0-3.2 KB	57-73 B
Permuted Kernel	PERK v2.1	3.5 KB	100 B
Rank Syndrome Decoding	RYDE v2	3.1 KB	69 B

Round-3

Round-2

Using seed trees of around 2048 leaves

🧐 ***What about the computational cost?***

*When using **seed trees** to commit to polynomials, we have*

Signing time ≈ Verification time ≈ few milliseconds (1-10 Mc)

over laptop-grade CPU

Comparison with the other families

	Lattice-based schemes				*UOV-like schemes*			*Alternative code-based schemes*	
	MPCitH	***Dilithium ML-DSA***	***Falcon FN-DSA***	***SPHINCS⁺ SLH-DSA***	***UOV***	***Mayo***	***SQIsign***	***LESS***	***CROSS***
Type	FS	FS	H&S	Hash-based	H&S	H&S	FS	FS	FS
|Sig|	2.5-4.5	2.4	0.7	7.8-17	0.1	0.2-0.5	0.1	1.3-2.3	9.0-18
|PK|	< 0.2	1.3	0.9	< 0.1	44-67	1.4-4.9	0.1	14-97	0.1
|Sig|+|PK|	2.5-4.6	3.7	1.6	7.9-17	44-67	1.9-5.1	0.2	17-98	9.0-18
Sign. Time	~	++	++	--	~	~	-	-	+
Verif. Time	~	++	++	~	++	++	~	-	+
Security	AES Unstructured SD Unstructured MQ …	Structured Lattice	Structured Lattice	Hash	UOV Trapdoor	New UOV-like Trapdoor	Isogeny	Code Equivalence	Restricted Syndrome Decoding

Sizes in kilobytes (KB)

FS: Fiat-Shamir transformation
H&S: Hash-and-sign scheme

Conclusion

- MPC-in-the-Head
  - Very versatile and tunable
  - Can be applied to any PQ hardness assumption
  - A practical tool to build conservative signature schemes

Conclusion

- MPC-in-the-Head
  - Very versatile and tunable
  - Can be applied to any PQ hardness assumption
  - A practical tool to build conservative signature schemes

- PIOP-based MPCitH Frameworks
  - Lead to signatures of 2.5-5 kilobytes
  - Speed up the MPCitH schemes
  - Used in the lastest version of all NIST candidates

Current Research

- The design for hash-based zero-knowledge proofs of knowledge dedicated to standard signatures is ***stabilizing***, but the design for zero-knowledge proofs for small-to-medium statements is still improving.
  - Advanced signature schemes, anonymous credentials
  - ZKPoK for private keys, ciphertexts, signatures, …
  - ZKPoK for well-formness in MPC

Current Research

- The design for hash-based zero-knowledge proofs of knowledge dedicated to standard signatures is ***stabilizing***, but the design for zero-knowledge proofs for small-to-medium statements is still improving.
  - Advanced signature schemes, anonymous credentials
  - ZKPoK for private keys, ciphertexts, signatures, …
  - ZKPoK for well-formness in MPC

- The MPCitH paradigm can be derived to have generic frameworks for building efficient advanced signatures from any assumption.
  - Ring signature schemes from any computational assumptions

[FR23] Feneuil, Rivain: “Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments” (JoC 2025)

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni: “Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures” (ePrint 2024)

Current Research

- The design for hash-based zero-knowledge proofs of knowledge dedicated to standard signatures is ***stabilizing***, but the design for zero-knowledge proofs for small-to-medium statements is still improving.
  - Advanced signature schemes, anonymous credentials
  - ZKPoK for private keys, ciphertexts, signatures, …
  - ZKPoK for well-formness in MPC

- The MPCitH paradigm can be derived to have generic frameworks for building efficient advanced signatures from any assumption.
  - Ring signature schemes from any computational assumptions
  - Bling signature schemes from any hash-and-sign scheme and a commitment scheme.

[BFM+26] Bouillaguet, Feneuil, Maire, Rivain, Sauvage, Vergnaud: “Blinding Post-Quantum Hash-and-Sign Signatures” (S&P 2026)

[BFM+25] Bouillaguet, Feneuil, Maire, Rivain, Sauvage, Vergnaud: “Multivariate Commitments and Signatures with Efficient Protocols” (ePrint 2025/2035)

[BBB+26] Baum, Beckmann, Beullens, Mukherjee, Rechberger: “Multivariate Commitments and Signatures with Efficient Protocols” (ePrint 2026/109)

Current Research

- The design for hash-based zero-knowledge proofs of knowledge dedicated to standard signatures is ***stabilizing***, but the design for zero-knowledge proofs for small-to-medium statements is still improving.
  - Advanced signature schemes, anonymous credentials
  - ZKPoK for private keys, ciphertexts, signatures, …
  - ZKPoK for well-formness in MPC

- The MPCitH paradigm can be derived to have generic frameworks for building efficient advanced signatures from any assumption.
  - Ring signature schemes from any computational assumptions
  - Bling signature schemes from any hash-and-sign scheme and a commitment scheme.
  - Threshold signatures from any computational assumptions

    [FRVW26] Feneuil, Rivain, Vergnaud, Warmé-Janville: “Threshold Signatures in the Head” (ePrint 2026/1125)

Current Research

- The design for hash-based zero-knowledge proofs of knowledge dedicated to standard signatures is ***stabilizing***, but the design for zero-knowledge proofs for small-to-medium statements is still improving.
  - Advanced signature schemes, anonymous credentials
  - ZKPoK for private keys, ciphertexts, signatures, …
  - ZKPoK for well-formness in MPC
- The MPCitH paradigm can be derived to have generic frameworks for building efficient advanced signatures from any assumption.
  - Ring signature schemes from any computational assumptions
  - Bling signature schemes from any hash-and-sign scheme and a commitment scheme.
  - Threshold signatures from any computational assumptions

Thank you for your attention.