MQOM: MQ on my Mind
— Version 2 —

Ryad Benadjila, Charles Bouillaguet
Thibauld Feneuil, Matthieu Rivain

PEPR PQ TLS

March 13, 2025, *Inria Paris*

CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS

- Round-2 Updates for MPCitH-based schemes
- High-level idea of MQOM v2
- Benchmarks of MQOM v2
- Conclusion

- 6 MPCitH-based schemes have been selected for round 2:

  FAEST, Mirath, MQOM, PERK, RYDE, SDitH

- Two new MPCitH frameworks since the previous NIST deadline:

  VOLE-in-the-Head (summer 2023) and **TC-in-the-Head** (fall 2023)

- 6 MPCitH-based schemes have been selected for round 2:

  FAEST, Mirath, MQOM, PERK, RYDE, SDitH

- Two new MPCitH frameworks since the previous NIST deadline:

  VOLE-in-the-Head (summer 2023) and **TC-in-the-Head** (fall 2023)

- Round-1 **FAEST** was relying on the **VOLEitH framework**, still the case for the round-2 version.

- Round-2 **SDitH** now relies on the **VOLEitH framework**.

- Round-2 versions of **Mirath**, **MQOM**, and **RYDE** now rely on the **TCitH framework**.

- Round-1 **PERK** was relying on the **shared-permutation framework,** still the case for the round-2 version.

- The both frameworks are **interchangeable**, several schemes mention a variant with the other framework.

- The both frameworks are **interchangeable**, several schemes mention a variant with the other framework.

- Mirath, MQOM, RYDE and SDitH uses the **PIOP formalism** to describe their scheme, instead of a sharing-based formalism (TCitH) or a VOLE-based formalism (VOLEitH).

- The both frameworks are **interchangeable**, several schemes mention a variant with the other framework.

- Mirath, MQOM, RYDE and SDitH uses the **PIOP formalism** to describe their scheme, instead of a sharing-based formalism (TCitH) or a VOLE-based formalism (VOLEitH).

- FAEST, Mirath, MQOM, RYDE and SDitH now primarily utilize **Rijndael-based** (AES-128, …) as symmetric primitives (for pseudorandom generator and commitment), shifting away from Keccak-based hashes to improve the scheme's speed.

- The both frameworks are **interchangeable**, several schemes mention a variant with the other framework.

- Mirath, MQOM, RYDE and SDitH uses the **PIOP formalism** to describe their scheme, instead of a sharing-based formalism (TCitH) or a VOLE-based formalism (VOLEitH).

- FAEST, Mirath, MQOM, RYDE and SDitH now primarily utilize **Rijndael-based** (AES-128, …) as symmetric primitives (for pseudorandom generator and commitment), shifting away from Keccak-based hashes to improve the scheme's speed.

- MQOM and SDitH use only **binary fields**, moving away from prime fields.

- The both frameworks are **interchangeable**, several schemes mention a variant with the other framework.

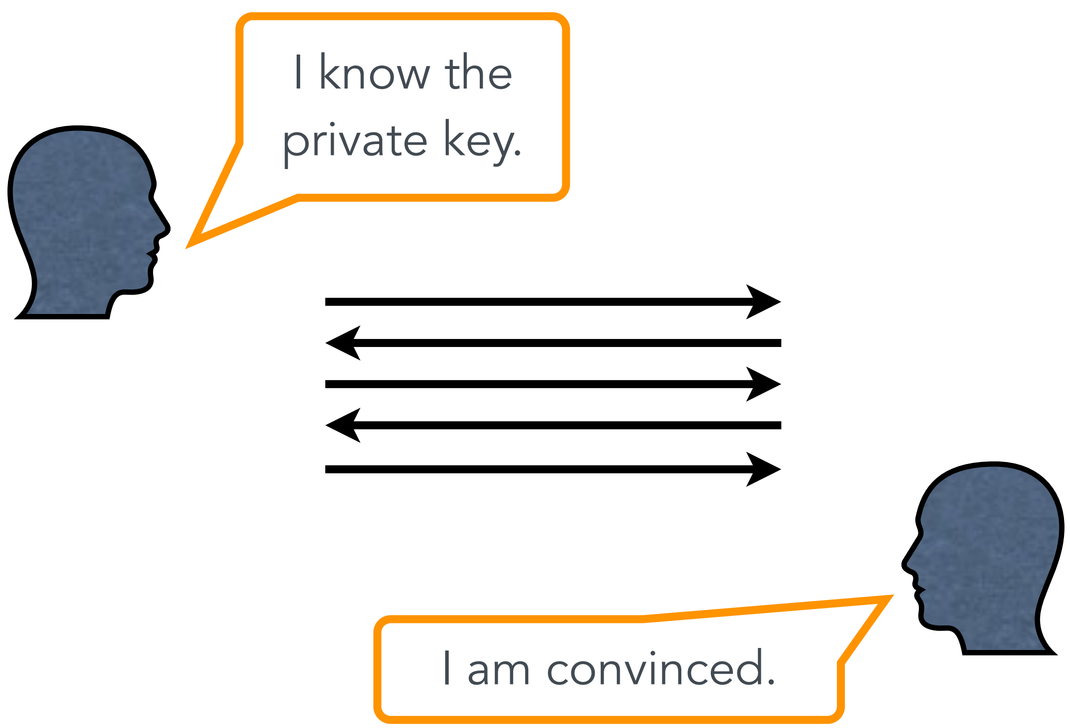
- Mirath, MQOM, RYDE and SDitH uses the **PIOP formalism** to describe their scheme, instead of a sharing-based formalism (TCitH) or a VOLE-based formalism (VOLEitH).

- FAEST, Mirath, MQOM, RYDE and SDitH now primarily utilize **Rijndael-based** (AES-128, …) as symmetric primitives (for pseudorandom generator and commitment), shifting away from Keccak-based hashes to improve the scheme's speed.

- MQOM and SDitH use only **binary fields**, moving away from prime fields.

- While the round-1 versions of those schemes have sizes between 5.5 KB and 10.5 KB for the first security level, the round-2 versions have sizes **between 2.8 KB and 5.9 KB**, with keys of several hundred bytes.

From an identification scheme



Multivariate Quadratic Problem

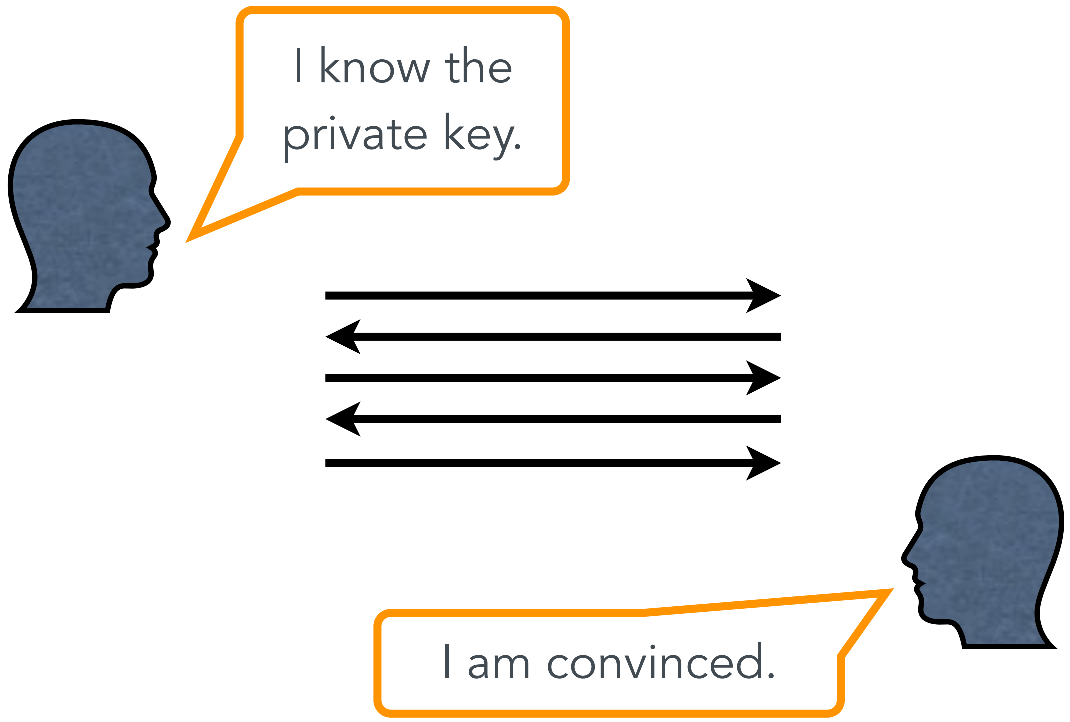
From m quadratic multivariate polynomials $f_1, \ldots, f_m$, find $x_1, \ldots, x_n \in \mathbb{F}_q$ such that

$$\begin{cases} f_1(x_1, \ldots, x_n) & = 0, \\ & \vdots \\ f_m(x_1, \ldots, x_n) & = 0. \end{cases}$$

For example ($n = m = 2$), find $x, y \in \mathbb{F}_q$ such that

$$\begin{cases} x^2 - y^2 + 2x + 5 = 0 \\ 4x^2 - x - 3y - 1 = 0. \end{cases}$$

From an identification scheme



Multivariate Quadratic Problem

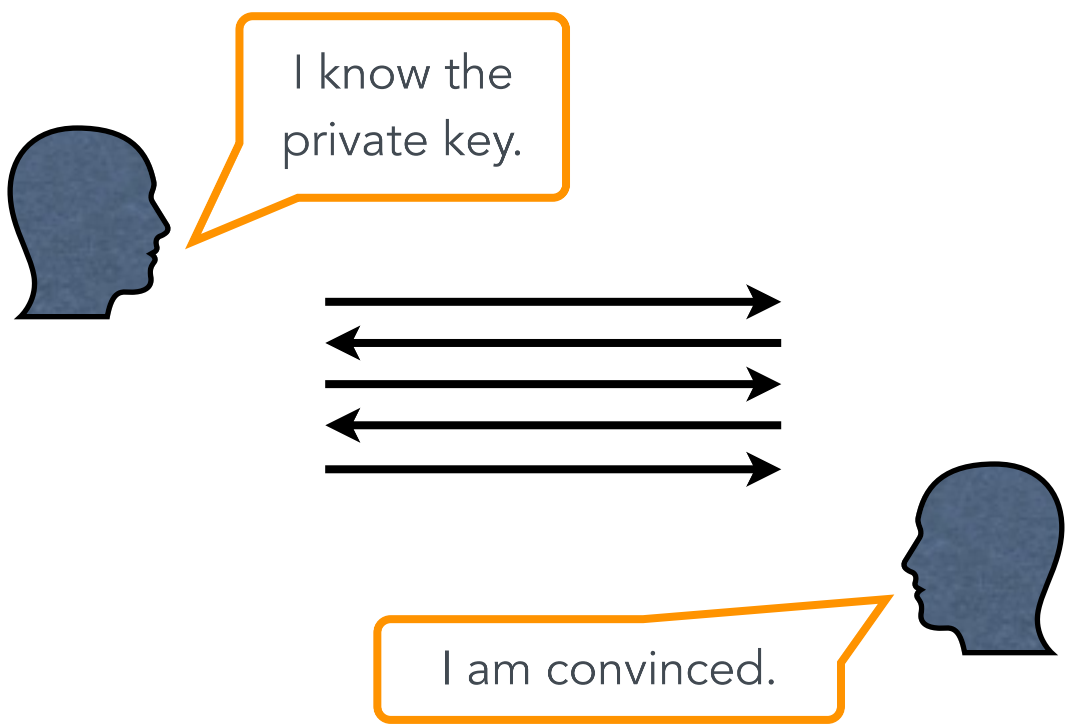
From m quadratic multivariate polynomials $f_1, \ldots, f_m$, find $x_1, \ldots, x_n \in \mathbb{F}_q$ such that

$$\begin{cases} f_1(x_1, \ldots, x_n) & = 0, \\ & \vdots \\ f_m(x_1, \ldots, x_n) & = 0. \end{cases}$$

Public Key: a random multivariate quadratic system $(f_1, \ldots, f_m)$

Secret Key: the MQ solution $x_1, \ldots, x_n$

From an identification scheme



Multivariate Quadratic Problem

From m quadratic multivariate polynomials $f_1, \ldots, f_m$, find $x_1, \ldots, x_n \in \mathbb{F}_q$ such that

$$\begin{cases} f_1(x_1, \ldots, x_n) & = 0, \\ & \vdots \\ f_m(x_1, \ldots, x_n) & = 0. \end{cases}$$

Public Key: a random multivariate quadratic system $(f_1, \ldots, f_m)$

Secret Key: the MQ solution $x_1, \ldots, x_n$

Used parameters: $n = m$, *over the field* $\mathbb{F}_2$ *or* $\mathbb{F}_{256}$.

The TCitH and VOLEitH frameworks can be described with the PIOP formalism.

- Manipulated objects in TCitH: **(Shamir's secret) sharings**
- Manipulated objects in VOLEitH: **VOLE correlations**
- Manipulated objects in PIOP: **Polynomials**

Lead to a description that **does not depend on MPC technologies**, leading to an **easier-to-understand** scheme for those who do not already know those two frameworks

The TCitH and VOLEitH frameworks can be described with the PIOP formalism.

- Manipulated objects in TCitH: **(Shamir's secret) sharings**
- Manipulated objects in VOLEitH: **VOLE correlations**
- Manipulated objects in PIOP: **Polynomials**

Lead to a description that **does not depend on MPC technologies**, leading to an **easier-to-understand** scheme for those who do not already know those two frameworks

For more details, see the talk:

Feneuil. ***The Polynomial-IOP Vision of the Latest MPCitH Frameworks for Signature Schemes***. *Post-Quantum Algebraic Cryptography - Workshop 2, IHP. 2024-11-08. Recording available online.*

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

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

where f is a public **degree-**2 **polynomial**.

Prove it!

Prover

Verifier

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

Sample a random degree-1 polynomial $P_0(X)$

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

$\text{Com}(P_0, P_1, \ldots, P_n)$ →

Prover

Verifier

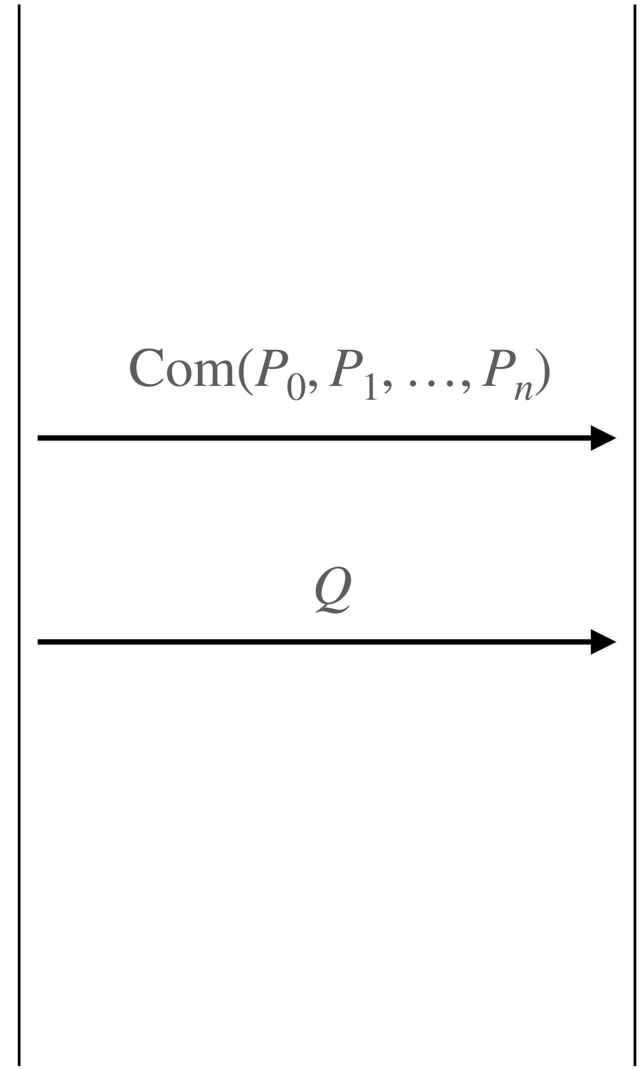
① For all i, sample a random degree-1 polynomial $P_i(X)$ such that $P_i(0) = w_i$

Sample a random degree-1 polynomial $P_0(X)$

② Commit 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

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

Sample a random degree-1 polynomial $P_0(X)$

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

$\text{Com}(P_0, P_1, \ldots, P_n)$ →

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

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

Sample a random degree-1 polynomial $P_0(X)$

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

$\text{Com}(P_0, P_1, \ldots, P_n)$ →

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 a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

$v_0, v_1, \ldots, v_n$ →

Prover

Verifier

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

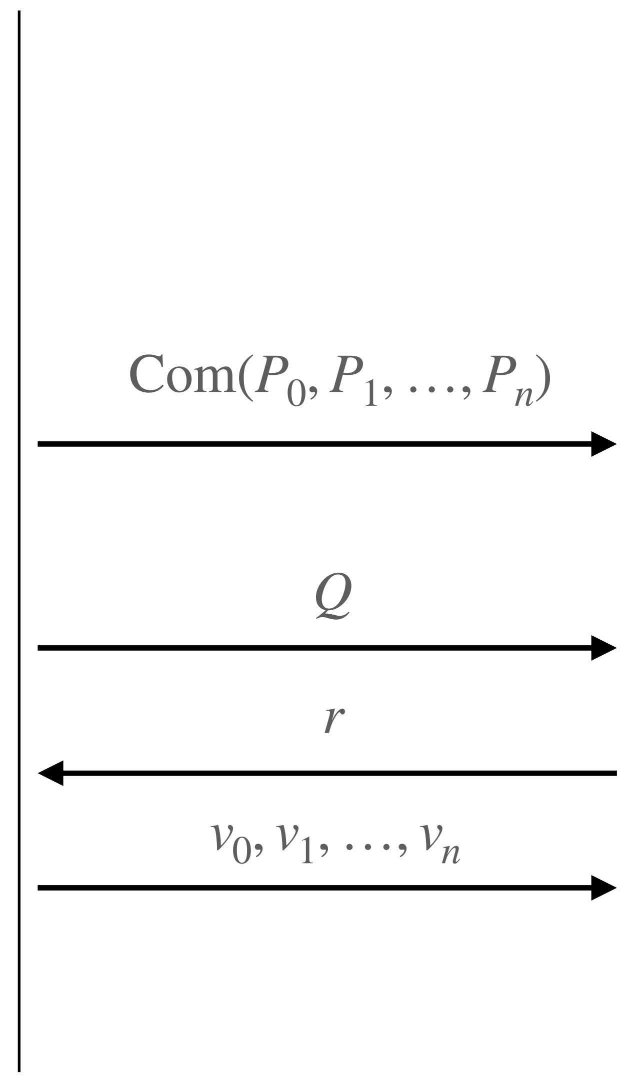
Sample a random degree-1 polynomial $P_0(X)$

② Commit the polynomials $P_0, P_1, \dots, P_n$

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

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

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.



④ Choose a random evaluation point $r \in S \subset \mathbb{F}$

⑥ Check that $v_0, v_1, \dots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + f(v_1, \dots, v_n)$$

Prover

Verifier

Soundness Analysis

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

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

Choose a degree-1 polynomial $P_0(X)$

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

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

③ Reveal the 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 a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

$v_0, v_1, \ldots, v_n$ →

⑥ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + f(v_1, \ldots, v_n)$$

Malicious Prover 😈

Verifier

Soundness Analysis

Malicious Prover 😈

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

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

Choose a degree-1 polynomial $P_0(X)$

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

③ Reveal the 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 Reveal the evaluation $v_i := P_i(r)$ for all i.

Evaluation into 0

$= 0$ $\neq 0$

Prover → Verifier: $\mathrm{Com}(P_0, P_1, \ldots, P_n)$

Prover → Verifier: Q

Verifier → Prover: r

Prover → Verifier: $v_0, v_1, \ldots, v_n$

Verifier

④ Choose a random evaluation point $r \in S \subset \mathbb{F}$

⑥ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + f(v_1, \ldots, v_n)$$

Soundness Analysis

③ Reveal the polynomial $Q(X)$. We know that $X \cdot Q(X) \neq X \cdot P_0(X) + f(P_1(X), \ldots, P_n(X))$

Choose a random evaluation point $r \in S \subset \mathbb{F}$

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

Check that $r \cdot Q(r) = r \cdot v_0 + f(v_1, \ldots, v_n)$

Malicious Prover 😈

Verifier

Soundness Analysis

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

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

Choose a random evaluation point $r \in S \subset \mathbb{F}$

Schwartz-Zippel Lemma: Let D be the **non-zero** degree-2 polynomial defined as

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

We have

$$\Pr[\text{verification passes}] = \Pr\left[D(r) = 0 \mid r \leftarrow_\$ S\right] \leq \frac{2}{|S|}.$$

Check that

$r \cdot Q(r) = r \cdot v_0 + f(v_1, \ldots, v_n)$

Verifier

Zero-Knowledge Analysis

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

Sample a random degree-1 polynomial $P_0(X)$

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

$\text{Com}(P_0, P_1, \ldots, P_n)$ →

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 a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

$v_0, v_1, \ldots, v_n$ →

⑥ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

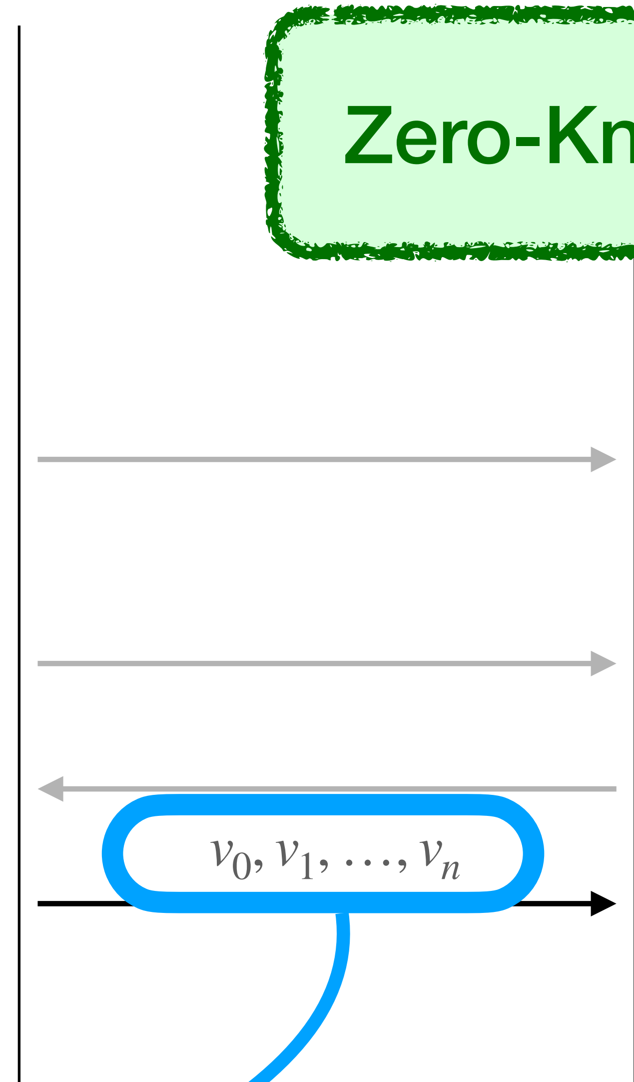
$$r \cdot Q(r) = r \cdot v_0 + f(v_1, \ldots, v_n)$$

Prover

Verifier 👀

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

Zero-Knowledge Analysis



Revealing an evaluation of $P_i(X)$ leaks no information about w_i.

Verifier 👀

Zero-Knowledge Analysis

Sample a random degree-1 polynomial $P_0(X)$

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

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

Verifier 👀

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

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

where f is a public **degree-2 polynomial**.

Prove it!

Prover

Verifier

Soundness Error = $\frac{2}{|S|}$

Probability that a malicious prover can convince the verifier.

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-**2 **polynomials**.

Prove it!

Prover

Verifier

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

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

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

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

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 a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

$\boldsymbol{v}_0, v_1, \ldots, v_n$ →

⑥ Check that $\boldsymbol{v}_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q_1(r) = r \cdot v_{0,1} + f_1(v_1, \ldots, v_n)$$
$$\vdots$$
$$r \cdot Q_m(r) = r \cdot v_{0,m} + f_m(v_1, \ldots, v_n)$$

Prover

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

Sample m random degree-1 polynomials $\boldsymbol{P}_0(X) = \big(P_{0,1}(X), \ldots, P_{0,m}(X)\big)$

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

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

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 a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑤ Reveal the evaluation $v_i := P_i(r)$ for all i.

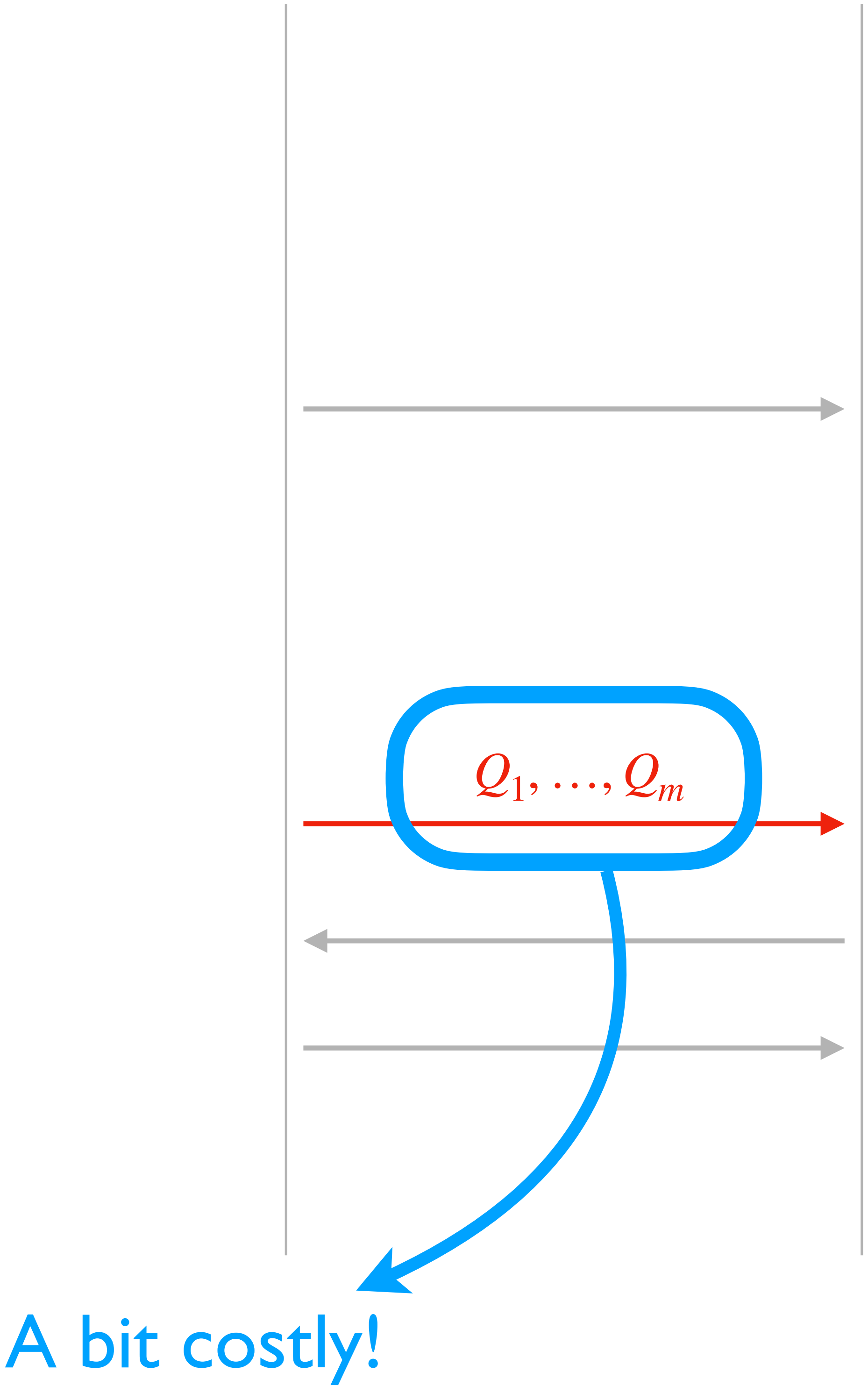
$\boldsymbol{v}_0, v_1, \ldots, v_n$ →

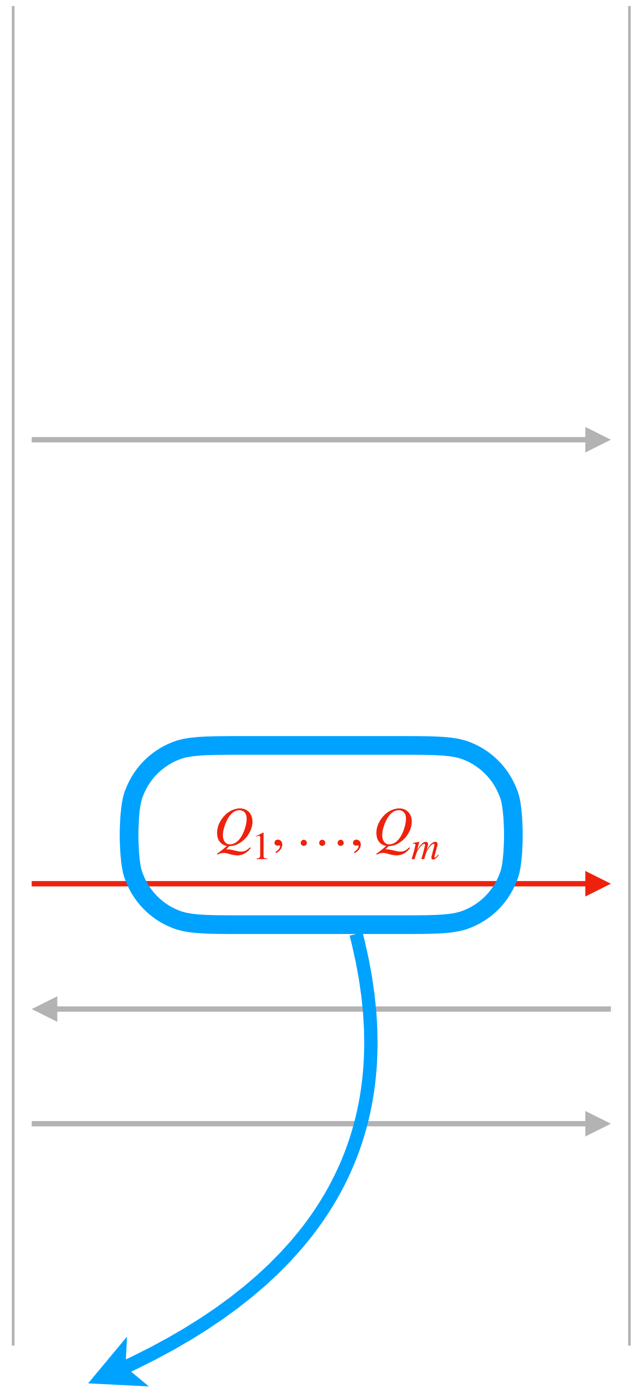
⑥ Check that $\boldsymbol{v}_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q_1(r) = r \cdot v_{0,1} + f_1(v_1, \ldots, v_n)$$
$$\vdots$$
$$r \cdot Q_m(r) = r \cdot v_{0,m} + f_m(v_1, \ldots, v_n)$$

Sigma variant (`3r`) of MQOM v2


$Q_1, \ldots, Q_m$
A bit costly!



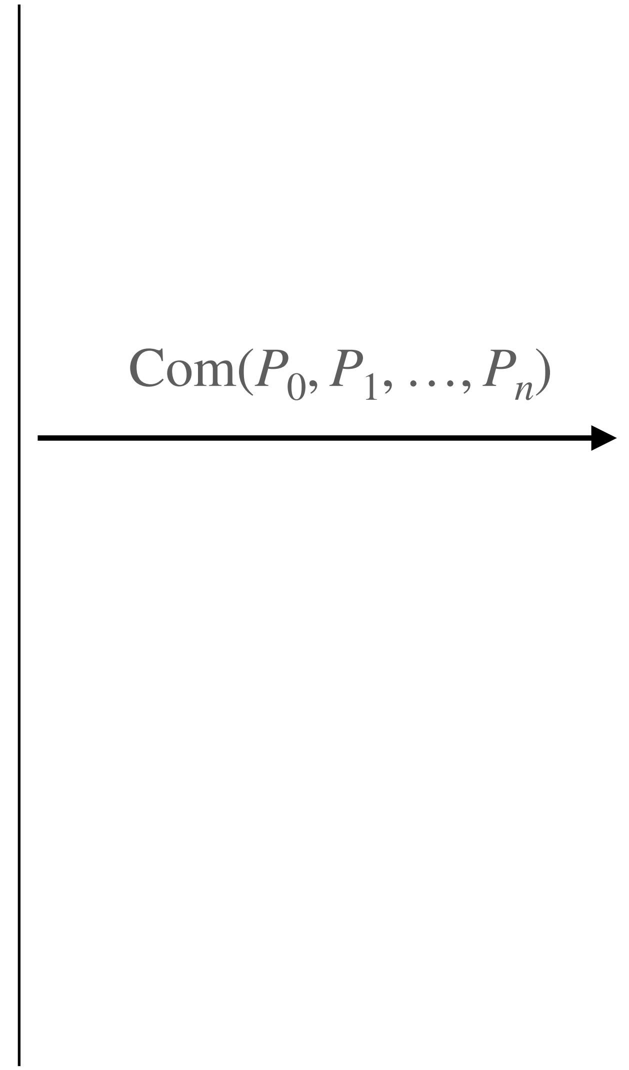
A bit costly!

Solution: batching

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

Sample a random degree-1 polynomial $P_0(X)$

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



Prover

Verifier

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

③ Choose random coefficients

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

← $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

Q →

Prover

Verifier

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

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

Well-defined!

$$\sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(0), \ldots, P_n(0)) = \sum_{j=1}^{m} \gamma_j \cdot f_j(w_1, \ldots, w_n)$$

$$= \sum_{j=1}^{m} \gamma_j \cdot 0 = 0$$

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

Sample a random degree-1 polynomial $P_0(X)$

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

Prover → Verifier: $\mathrm{Com}(P_0, P_1, \ldots, P_n)$

③ Choose random coefficients

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

Verifier → Prover: $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

Prover → Verifier: Q

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

Verifier → Prover: r

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

Prover → Verifier: $v_0, v_1, \ldots, v_n$

Prover

Verifier

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

③ Choose random coefficients

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

← $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

Q →

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

$v_0, v_1, \ldots, v_n$ →

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + \sum_{j=1}^{m} \gamma_j \cdot f_j(v_1, \ldots, v_n)$$

Prover

Verifier

Soundness Analysis

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

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

③ Choose random coefficients

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

← $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

→ Q

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

→ $v_0, v_1, \ldots, v_n$

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + \sum_{j=1}^{m} \gamma_j \cdot f_j(v_1, \ldots, v_n)$$

<u>Prover</u>　　　　　　<u>Verifier</u>

Soundness Analysis

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

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

Prover

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

← $\gamma_1, \ldots, \gamma_m$

Q →

← r

$v_0, v_1, \ldots, v_n$ →

③ Choose random coefficients

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

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the

$, v_n)$

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

$$\sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(0), \ldots, P_n(0)) \neq 0$$

Soundness Analysis

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

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

③ Choose random coefficients

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

← $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) \neq X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

→ Q

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

→ $v_0, v_1, \ldots, v_n$

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Check that

$$r \cdot Q(r) = r \cdot v_0 + \sum_{j=1}^{m} \gamma_j \cdot f_j(v_1, \ldots, v_n)$$

Schwartz-Zippel Lemma: Since it is a degree-2 relation,

$$\Pr[\text{verification passes}] \leq \frac{2}{|S|}.$$

Verifier

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

Sample a random degree-1 polynomial $P_0(X)$

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

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

③ Choose random coefficients

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

← $\gamma_1, \ldots, \gamma_m$

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

$$X \cdot Q(X) = X \cdot P_0(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X))$$

Q →

⑤ Choose a random evaluation point $r \in S \subset \mathbb{F}$

← r

⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

$v_0, v_1, \ldots, v_n$ →

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

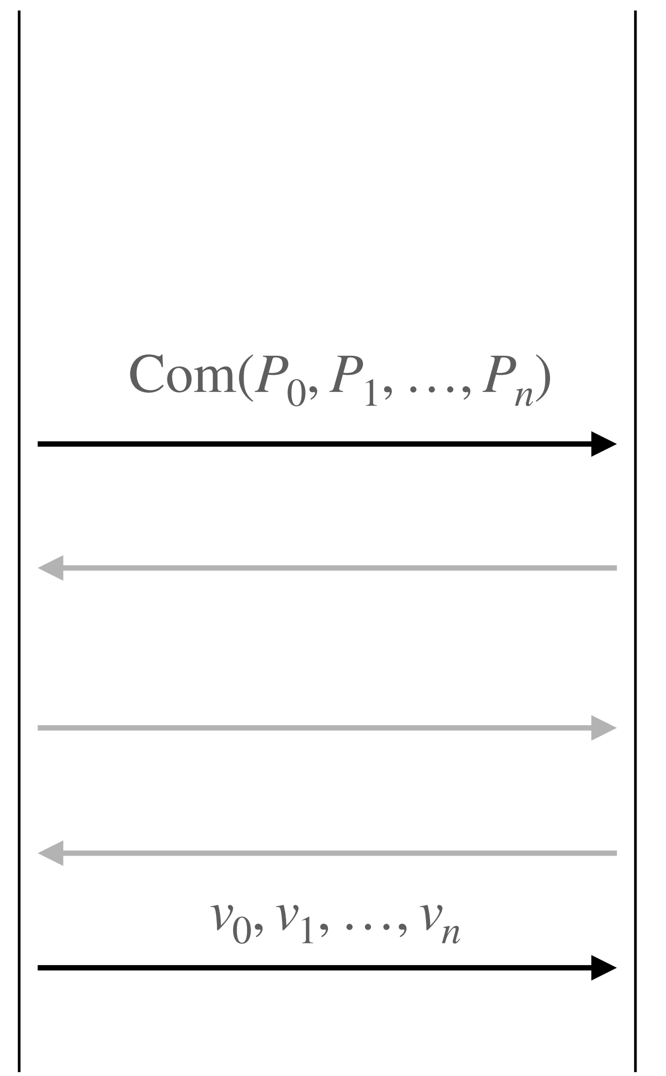
Check that

$$r \cdot Q(r) = r \cdot v_0 + \sum_{j=1}^{m} \gamma_j \cdot f_j(v_1, \ldots, v_n)$$

5-round variant (`5r`) of MQOM v2

<u>Verifier</u>

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

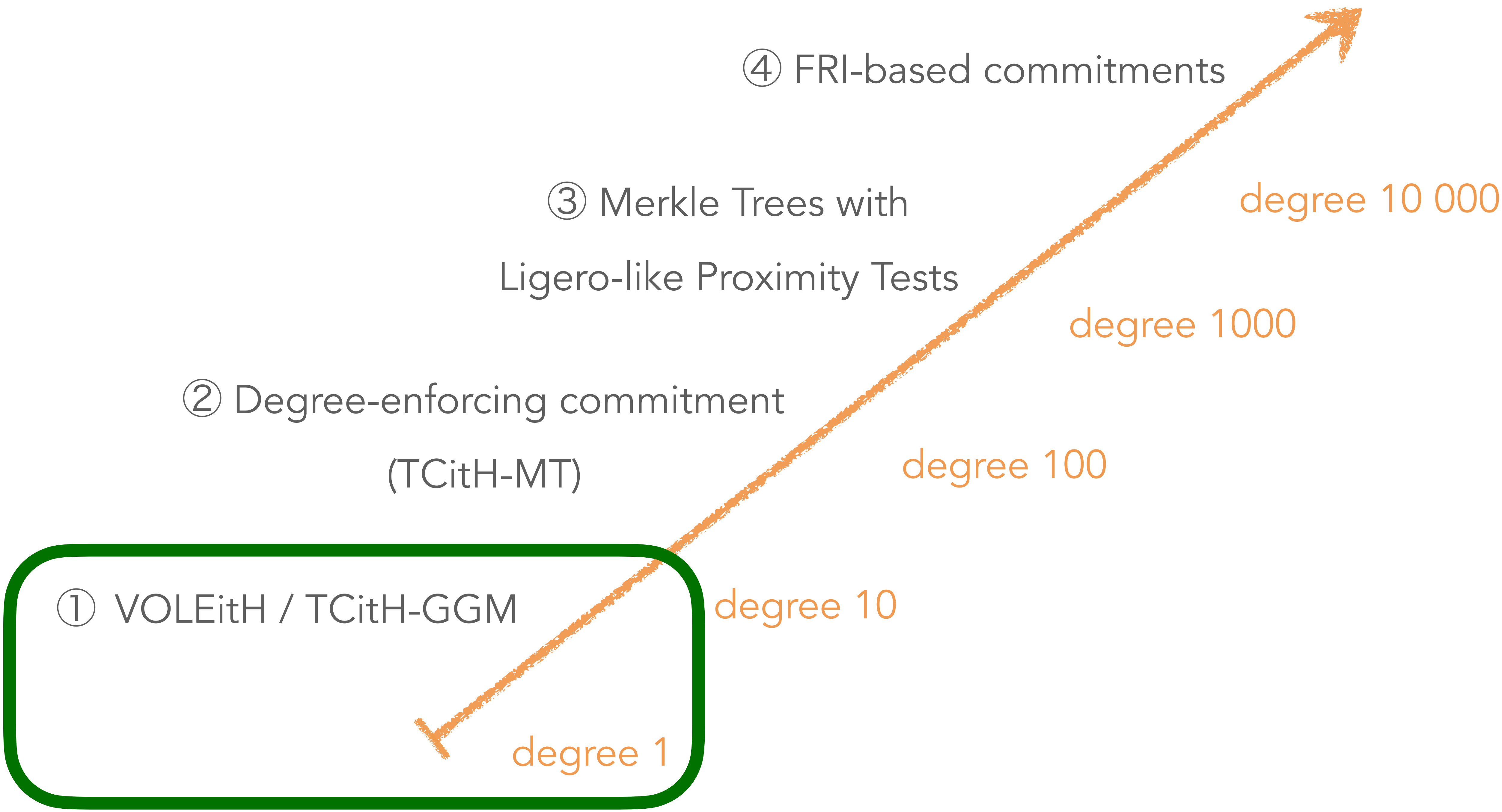


⑥ Reveal the evaluation $v_i := P_i(r)$ for all i.

⑦ Check that $v_0, v_1, \ldots, v_n$ are consistent with the commitment.

Prover

Verifier


④ FRI-based commitments
③ Merkle Trees with
Ligero-like Proximity Tests
② Degree-enforcing commitment
(TCitH-MT)
① VOLEitH / TCitH-GGM
degree 10 000
degree 1000
degree 100
degree 10
degree 1

Correctness:

If $N \geq 2$, P is a random degree-1 polynomial.

Correctness:
If $N \geq 2$, P is a random degree-1 polynomial.

Commitment:
We commit to each value r_i ***independently***.

Correctness:

If $N \geq 2$, P is a random degree-1 polynomial.

Commitment:

We commit to each value r_i ***independently***.

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

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

$$P(e_{i*}) = \sum_{i \neq i*} r_i \cdot R_i(e_{i*}) + r_{i*} \cdot \underbrace{R_{i*}(e_{i*})}_{=0}$$

$$= \sum_{i \neq i*} r_i \cdot R_i(e_{i*})$$

Correctness:

If $N \geq 2$, P is a random degree-1 polynomial.

Commitment:

We commit to each value r_i ***independently***.

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

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

$$P(e_{i^*}) = \sum_{i \neq i^*} r_i \cdot R_i(e_{i^*}) + r_{i^*} \cdot \underbrace{R_{i^*}(e_{i^*})}_{=0}$$

$$= \sum_{i \neq i^*} r_i \cdot R_i(e_{i^*})$$

The opening leaks *nothing* about P, except $P(e_{i^*})$.

Correctness:
If $N \geq 2$, P is a random degree-1 polynomial.

Commitment:
We commit to each value r_i ***independently***.

Opening $P(e_{i*})$:
Reveal all $\{r_i\}_{i \neq i*}$.

$$P(e_{i*}) = \sum_{i \neq i*} r_i \cdot R_i(e_{i*}) + r_{i*} \cdot \underbrace{R_{i*}(e_{i*})}_{=0}$$

$$= \sum_{i \neq i*} r_i \cdot R_i(e_{i*})$$

The opening leaks *nothing* about P, except $P(e_{i*})$.

🛠 Can be adapted to any degree.

Costly! 😰

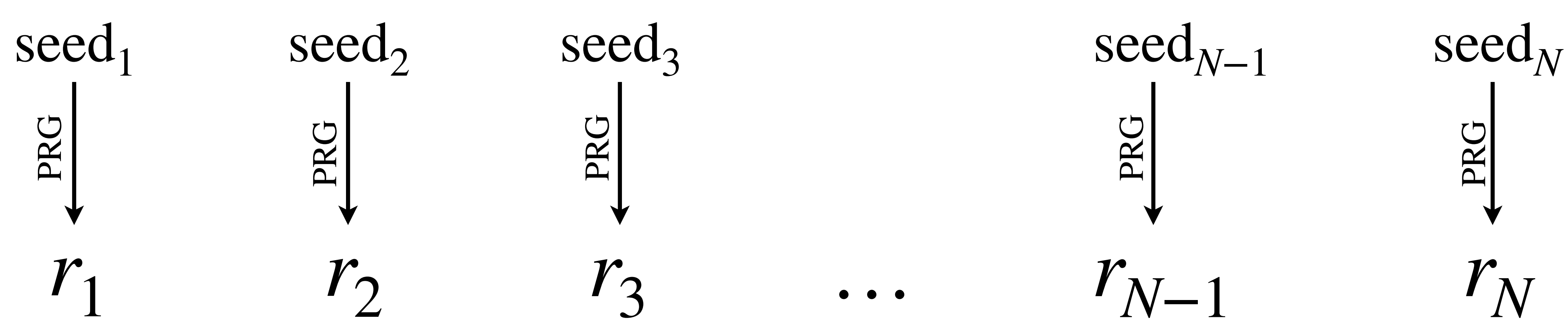
Opening $P(e_{i*})$:

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

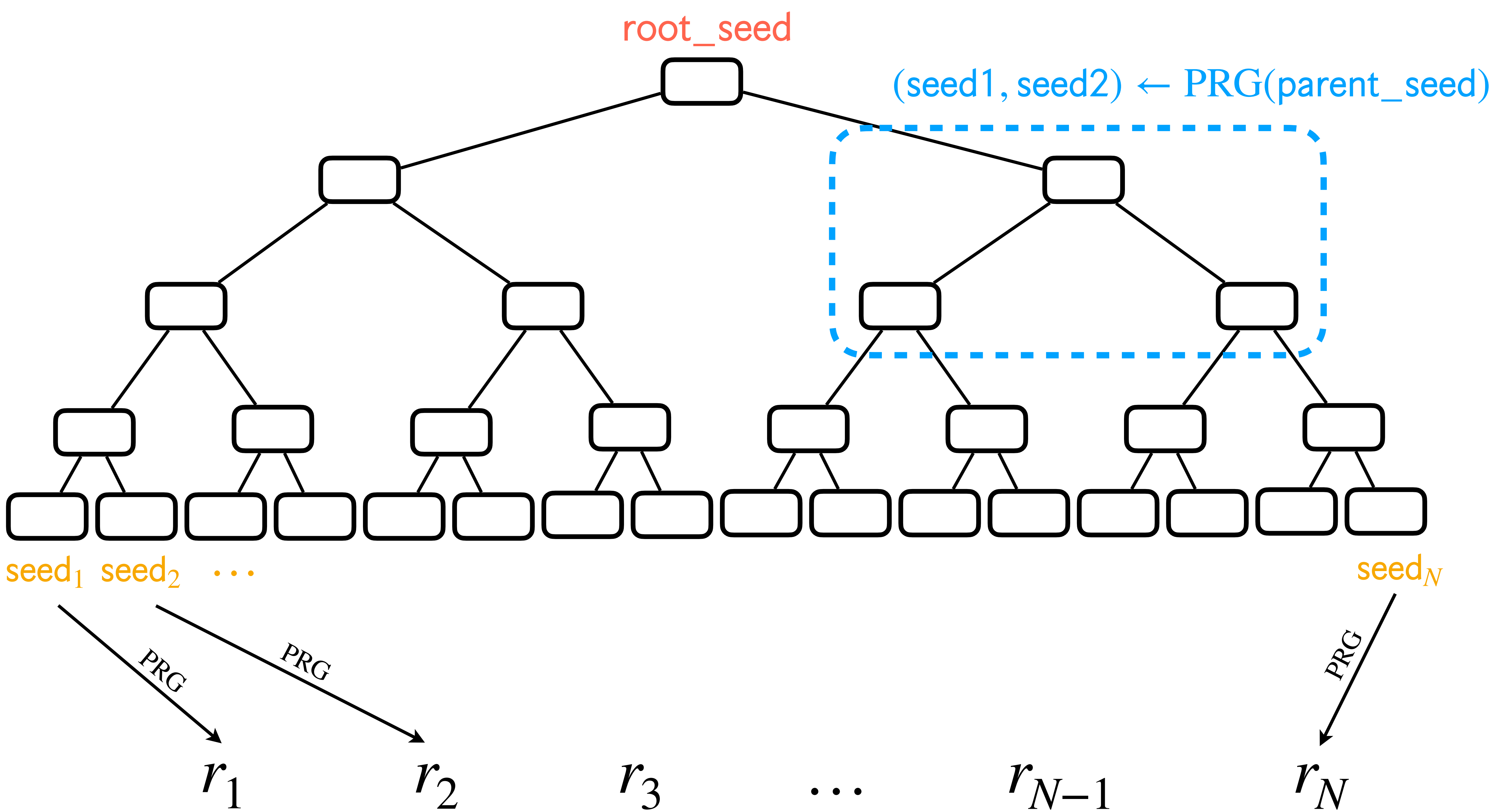
[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)

$$r_1 \quad r_2 \quad r_3 \quad \cdots \quad r_{N-1} \quad r_N$$

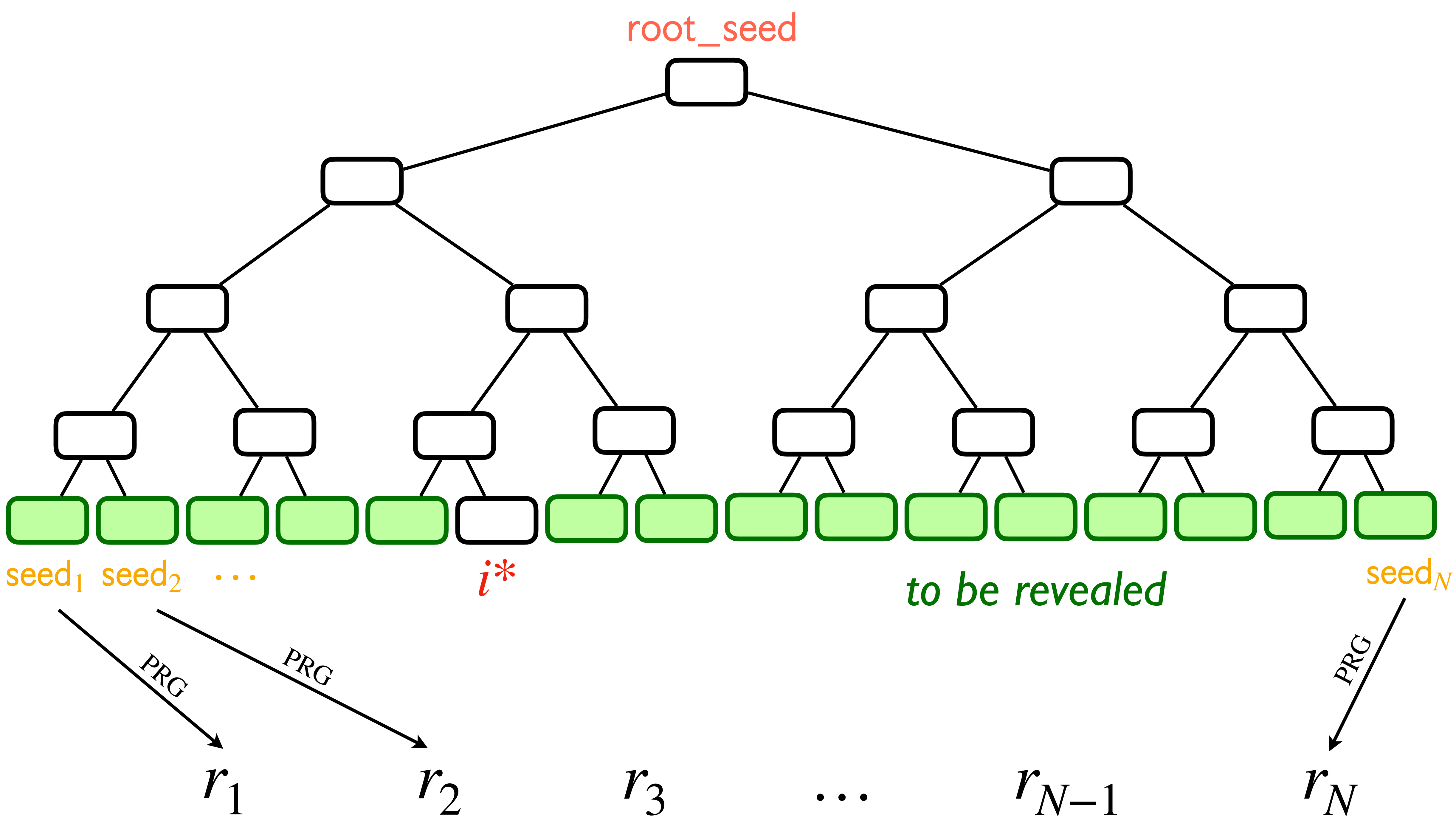
[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



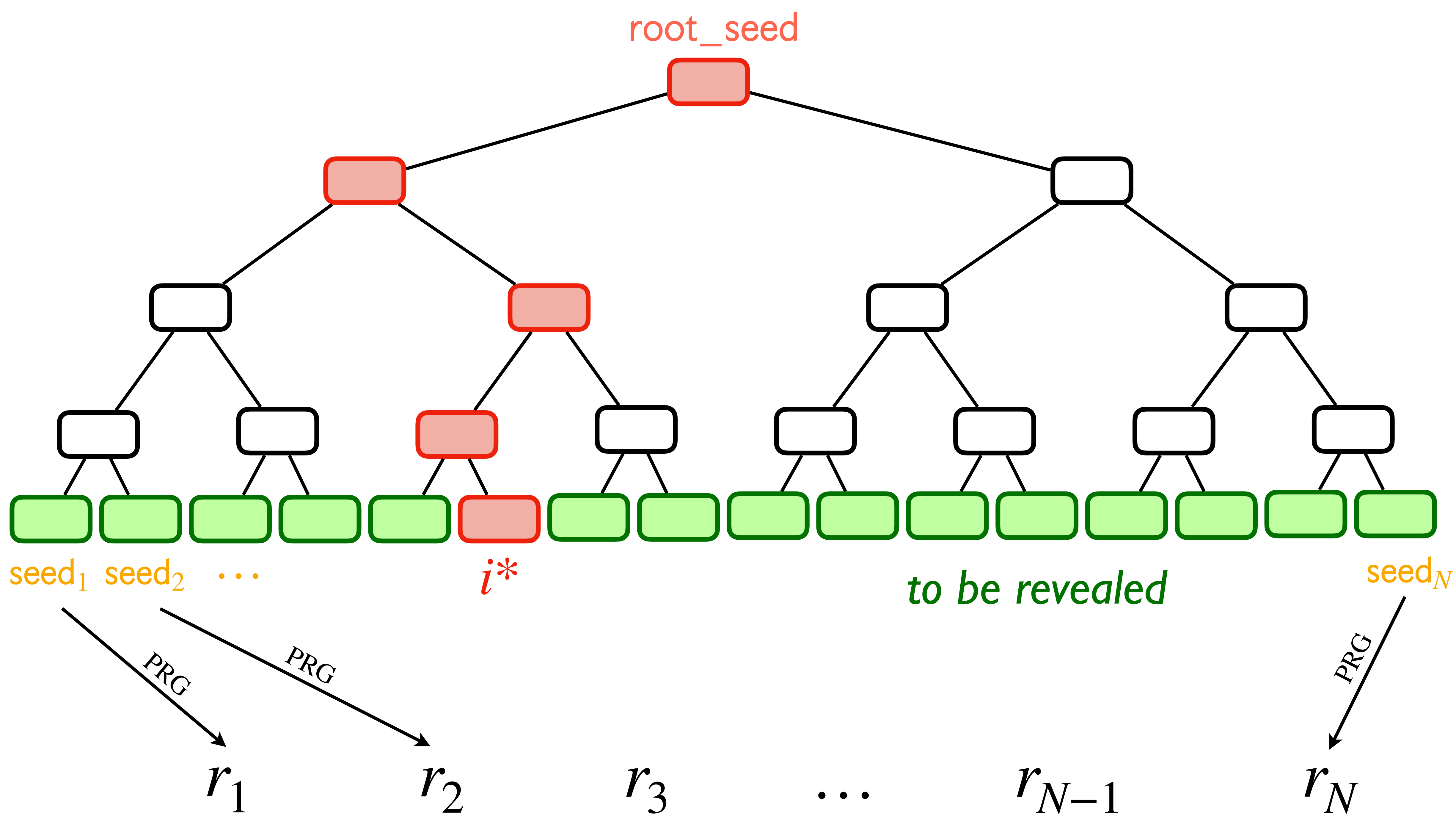
[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



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



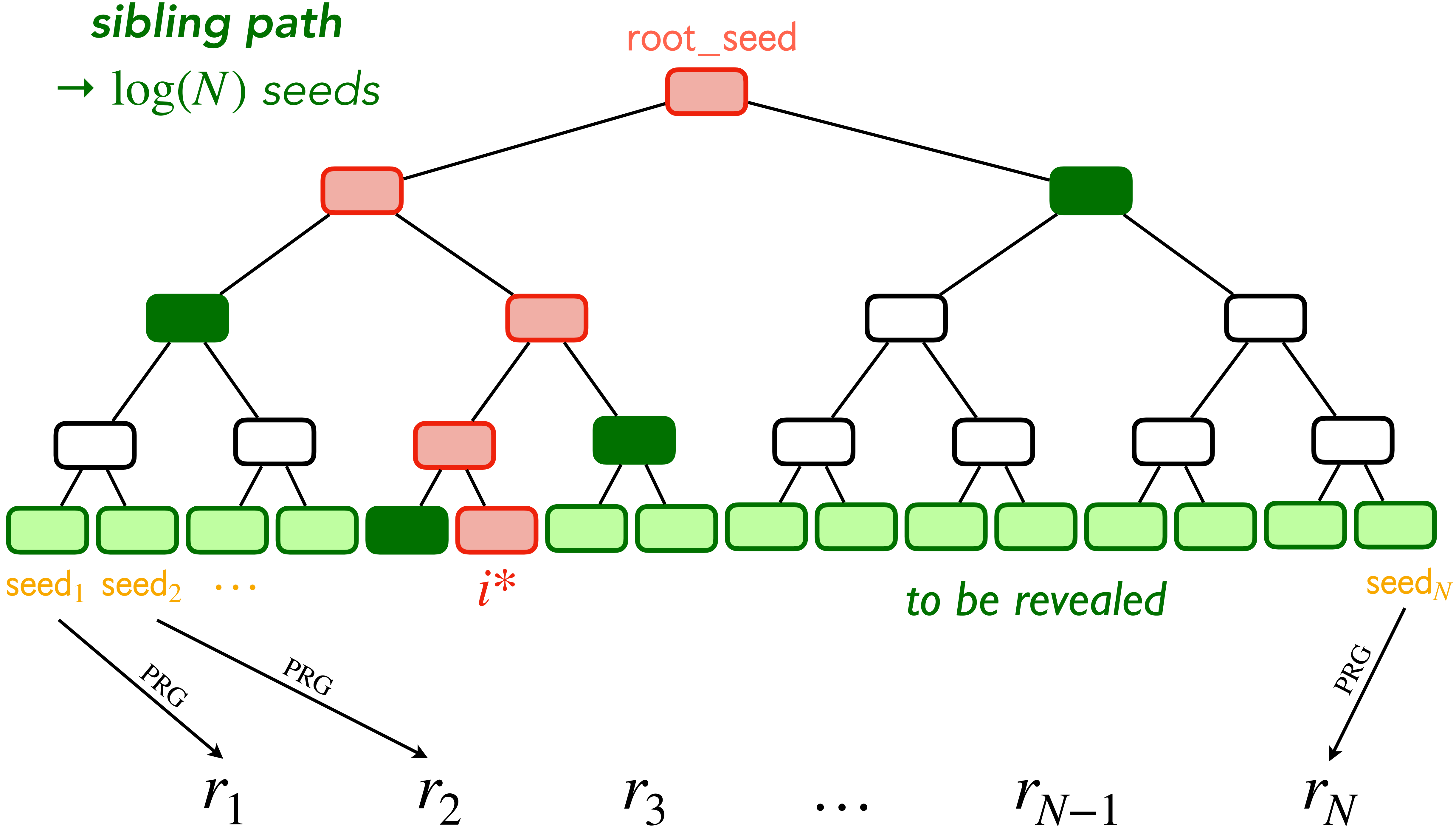
[GGM84] Goldreich, Goldwasser, Micali: "How to construct random functions (extended extract)" (FOCS 1984)



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

sibling path

→ $\log(N)$ *seeds*



Mirath
MQOM v2
RYDE v2

FAEST
SDitH

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-**2 **polynomials**.

Prove it!

Prover

Verifier

Fiat-Shamir Transformation

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-2 polynomials**.

Prover

Prove it!

Verifier

Signature Scheme

MQOMv2 Instance			*PK Size*	*Sizes (R3)*	*Sizes (R5)*	*Sig. / Verif. Running times*
NIST I	gf2	Short	52 B	**2 868 B**	*2 820 B*	≈ 18-20 Mcycles
		Fast		**3 212 B**	*3 144 B*	≈ 9-10 Mcycles
	gf256	Short	80 B	*3 540 B*	**3 156 B**	≈ 12-15 Mcycles
		Fast		*4 164 B*	**3 620 B**	≈ 3-4 Mcycles
NIST V	gf2	Short	104 B	**11 764 B**	*11 564 B*	≈ 133-143 Mcycles
		Fast		**13 412 B**	*13 124 B*	≈ 85-88 Mcycles
	gf256	Short	160 B	*14 564 B*	**12 964 B**	≈ 56-61 Mcycles
		Fast		*17 444 B*	**15 140 B**	≈ 14-15 Mcycles

		NIST Submission	
Security Assumptions		***Candidate Name***	***Sizes***
AES Block cipher	Secret Key	FAEST	4.5-5.9 KB
	Fixed Key (EM)	FAEST-EM	3.9-5.1 KB
MinRank	Field GF(2)	Mirath	2.9-3.5 KB
	Field GF(16)		3.1-3.7 KB
Multivariate Quadratic	Field GF(2)	MQOM	2.8-3.2 KB
	Field GF(256)		3.1-4.1 KB
Permuted Kernel	t=3	PERK	6.3-8.4 KB
	t=5		5.8-8.0 KB
Rank Syndrome Decoding		RYDE	3.0-3.6 KB
Syndrome Decoding		SDitH	3.7-4.5 KB

- Among the shortest MPCitH signature schemes:
  - Since all the other one-way functions as expressed as a **structured** (quadratic or cubic) multivariate system, it leads to **larger** systems for a given field, and so the MQ-based signature is the more efficient (in terms of communication).
- Among the simplest MPCitH signature schemes:
  - Do not need to arithmetize the one-way function as a multivariate system.
  - Rely on the TCitH framework
- MQOM v2 is the only NIST MPCitH-based candidate that has a variant with 3 rounds (the other schemes have 5 rounds or 7 rounds).

- Implementation effort for the new versions of the MPCitH-based schemes

- Fine-tuning of the parameters for trade-offs

- Many possible optimizations
    - Use of Rijndael-based ciphers (AES128, Rijndael-256-256, …) for seed derivation and seed commitments
    - Possible choices for tree derivation
    - …

- Implementation effort for the new versions of the MPCitH-based schemes

- Fine-tuning of the parameters for trade-offs

- Many possible optimizations
    - Use of Rijndael-based ciphers (AES128, Rijndael-256-256, …) for seed derivation and seed commitments
    - Possible choices for tree derivation
    - …

Thank you for your attention.