Special Soundness and Binding Properties: A Framework for Tightly Secure zk-SNARKs

Erki Külaots, Helger Lipmaa, Roberto Parisella, Janno Siim

Institute of Computer Science,
University of Tartu, Tartu, Estonia

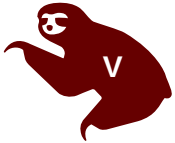
April 27, 2026

UNIVERSITY OF TARTU

Main goal

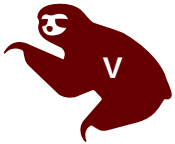
To provide a new proof technique for proving that a constructed SNARK is knowledge sound with a tight error bound.

Interactive Argument



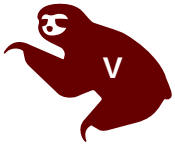
Interactive Argument

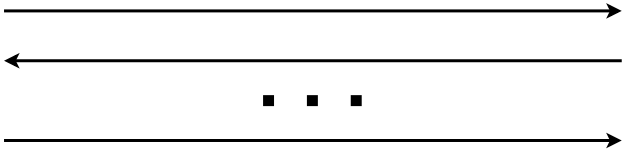
Statement x, witness w
x holds iff $\exists w : (x, w) \in R$



Interactive Argument

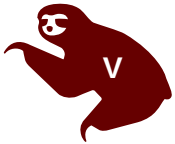
Statement x, witness w
x holds iff $\exists w : (x, w) \in R$

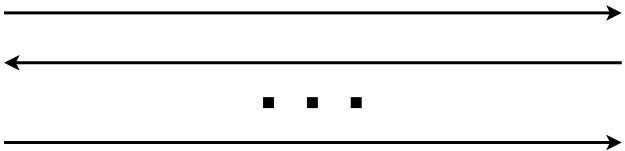




Interactive Argument

Statement x, witness w
x holds iff $\exists w : (x, w) \in R$

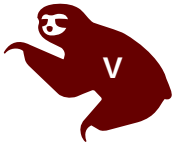




Accept or reject

Non-Interactive Argument

Statement x, witness w
x holds iff $\exists w : (x, w) \in R$

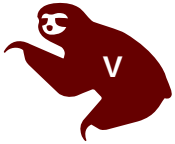


π

Accept or reject

Succinct Non-Interactive Argument of Knowledge (SNARK)

Statement x, witness w
x holds iff $\exists w : (x, w) \in R$



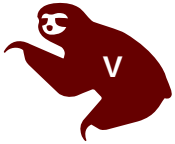
π

Accept or reject

- **Succinct** – π is small compared to w; verification is fast.

Succinct Non-Interactive Argument of Knowledge (SNARK)

Statement x, witness w
x holds iff $\exists w : (x, w) \in R$



π

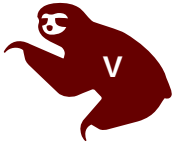
Accept or reject

- **Succinct** – π is small compared to w; verification is fast.
- One recipe for constructing SNARKs:

$$\text{SNARK} = \text{PIOP} + \text{PCS} + \text{Fiat Shamir transformation.}$$

Succinct Non-Interactive Argument of Knowledge (SNARK)

Statement x, witness w
x holds iff $\exists w : (x, w) \in R$



π

Accept or reject

- **Succinct** – π is small compared to w; verification is fast.
- One recipe for constructing SNARKs:

$$\text{SNARK} = \text{PIOP} + \text{PCS} + \text{Fiat Shamir transformation.}$$

Polynomial Interactive Oracle Proof (PIOP)

Challenge phase

Statement x, witness w

Polynomial Interactive Oracle Proof (PIOP)

Challenge phase

Statement x, witness w

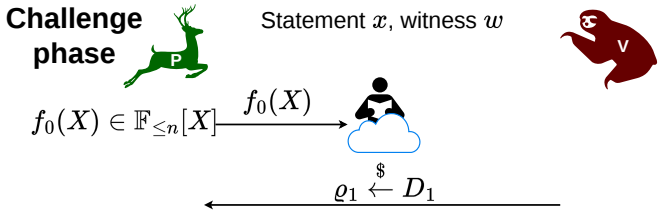
Polynomial Interactive Oracle Proof (PIOP)

Challenge phase

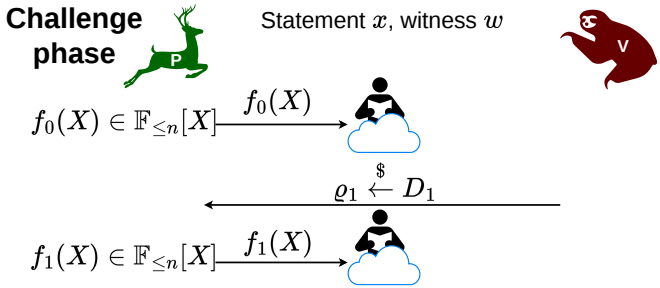
Statement x, witness w

$f_0(X) \in \mathbb{F}_{\leq n}[X]$ $\xrightarrow{f_0(X)}$

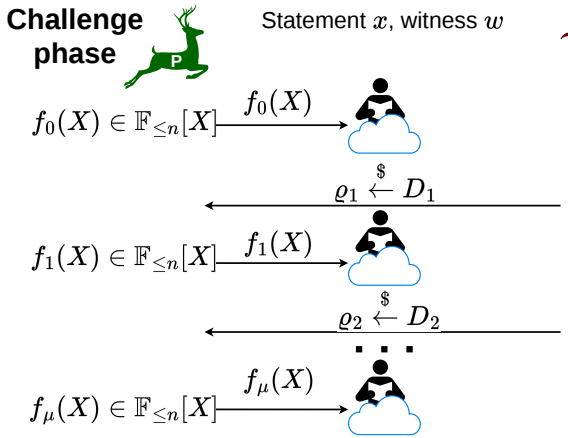
Polynomial Interactive Oracle Proof (PIOP)



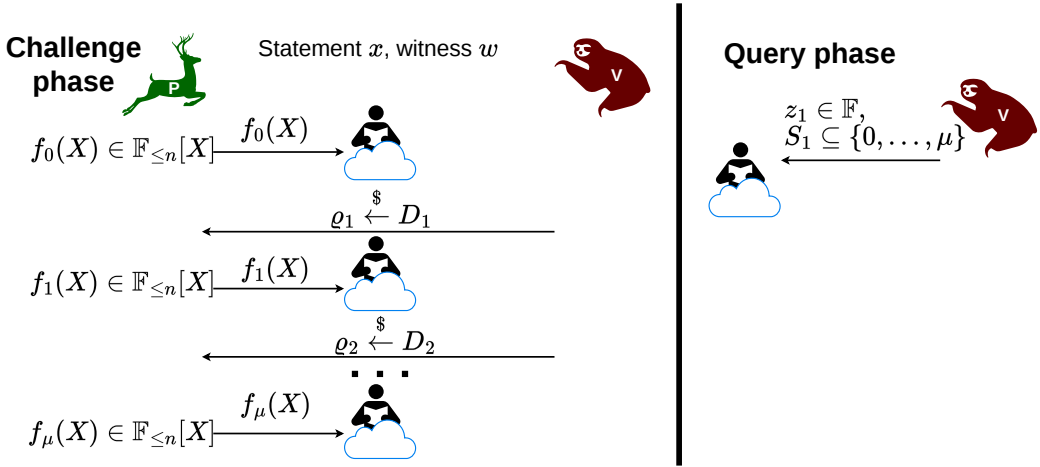
Polynomial Interactive Oracle Proof (PIOP)



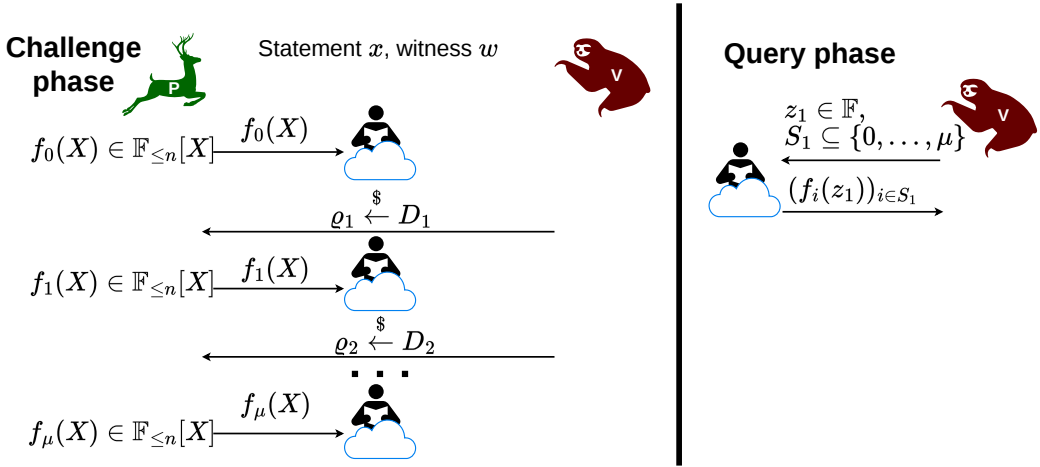
Polynomial Interactive Oracle Proof (PIOP)



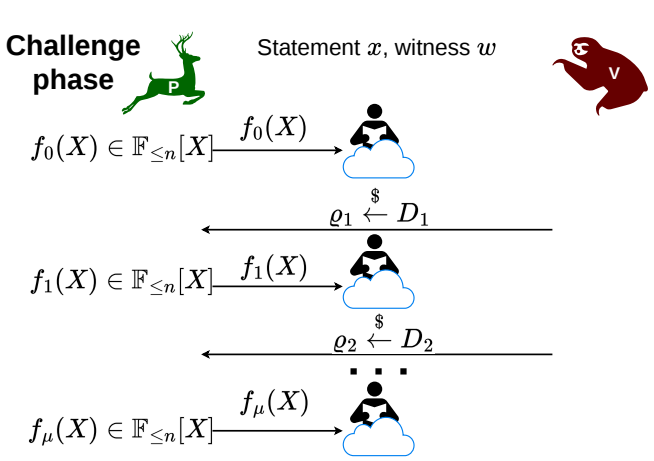
Polynomial Interactive Oracle Proof (PIOP)

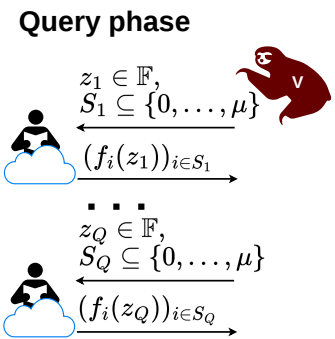


Polynomial Interactive Oracle Proof (PIOP)

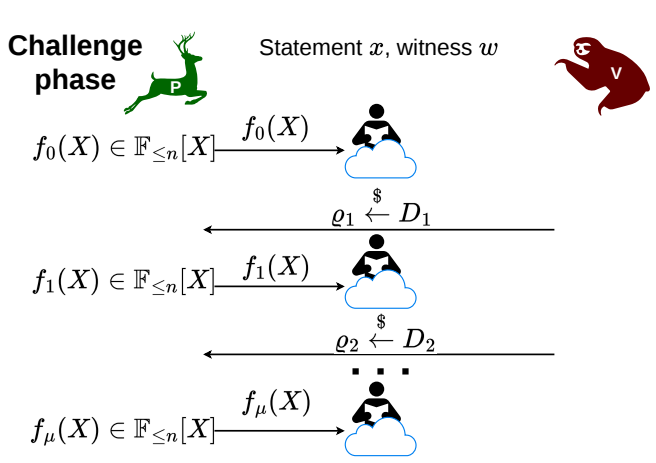


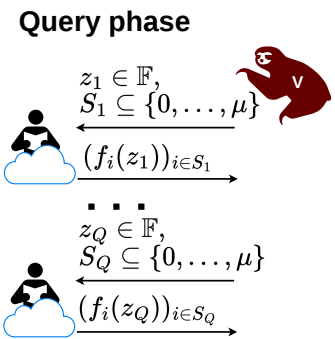
Polynomial Interactive Oracle Proof (PIOP)





Polynomial Interactive Oracle Proof (PIOP)





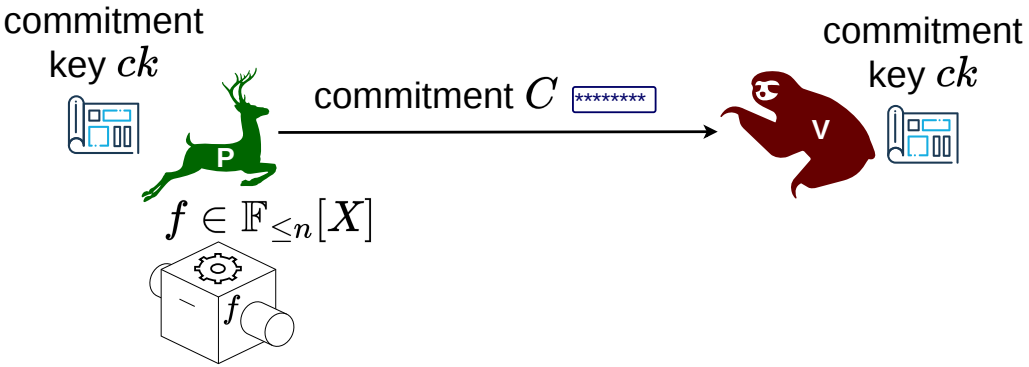
Requirements for the Oracle

We require from the oracle  to

1. accept polynomials with degree $\leq n$ in the challenge phase;
2. input a field element and a set of indices, and return the evaluations of the polynomials specified by the indices in the query phase.

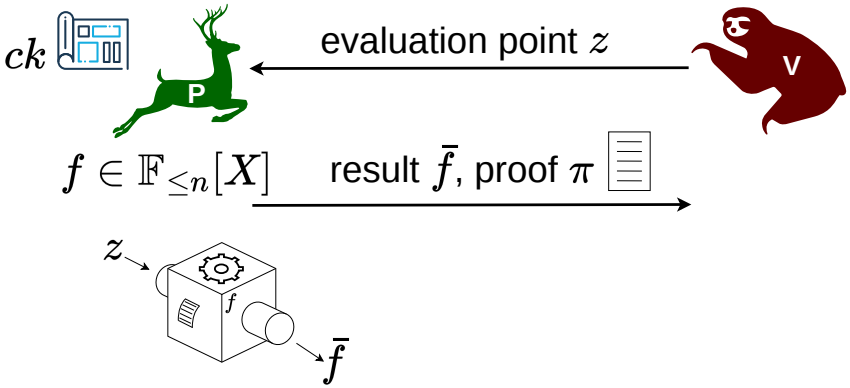
Polynomial Commitment Scheme (PCS)

Commitment



Polynomial Commitment Scheme (PCS)

Opening



Polynomial Commitment Scheme (PCS)

Verification

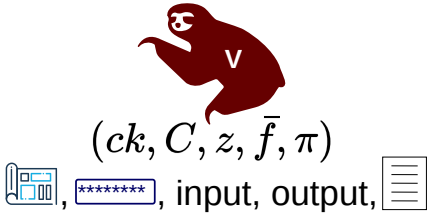
$(ck, C, z, \bar{f}, \pi)$


, ********, input, output,


Accept
or reject

Polynomial Commitment Scheme (PCS)

Verification



Note that there exist such commitment schemes. For example, see [KZG10].

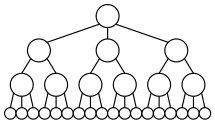
Soundness Notions for Interactive Arguments

- **Knowledge soundness** – There exists an extractor such that if a prover P can convince the verifier V with high enough probability that the statement holds, then with high probability the extractor can extract a witness w.

Soundness Notions for Interactive Arguments

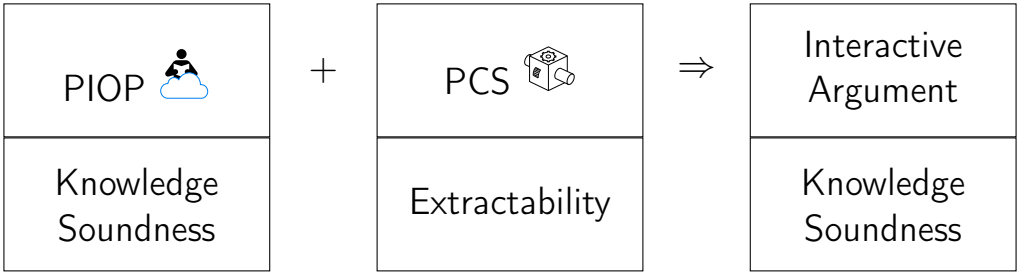
- **Knowledge soundness** – There exists an extractor such that if a prover P can convince the verifier V with high enough probability that the statement holds, then with high probability the extractor can extract a witness w.
- **Special soundness** – There exists an extractor such that, given a tree of accepting transcripts, it can output a witness w.

Tree of accepting transcripts → Extractor → witness w



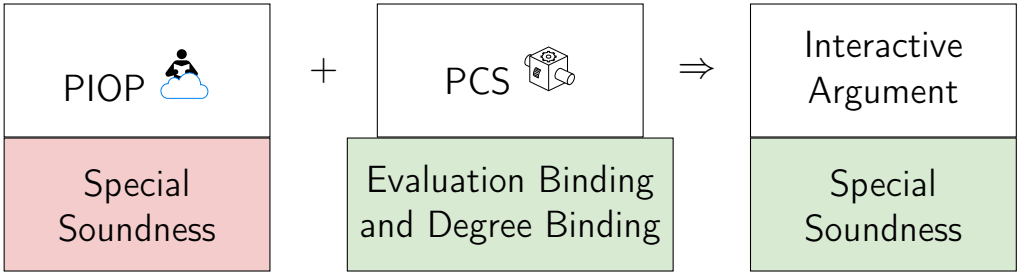
Motivation

Previous work



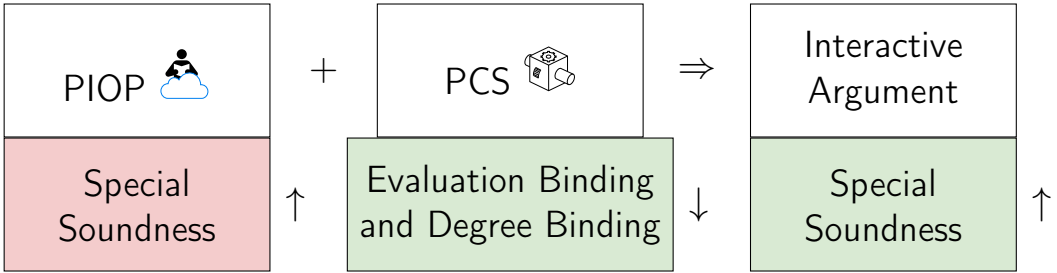
Motivation

Our work

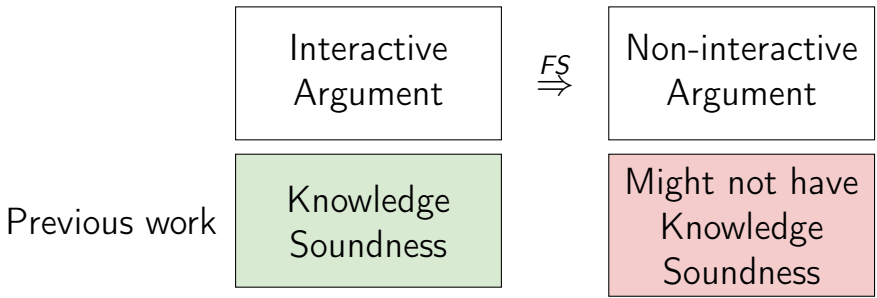


Motivation

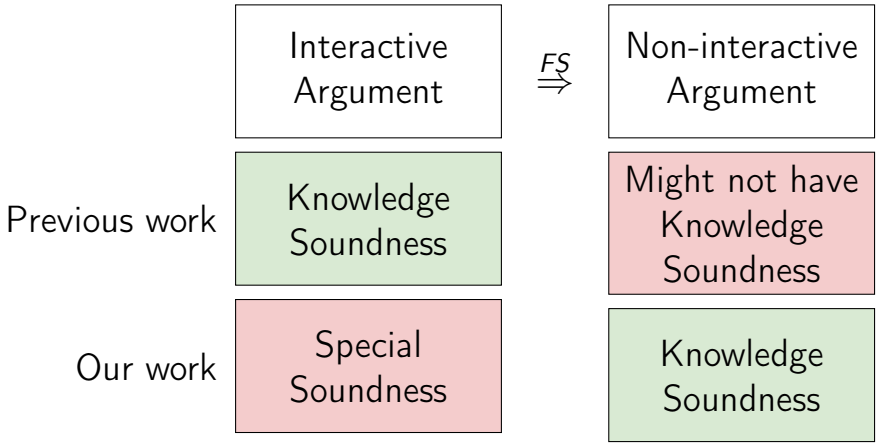
Our work



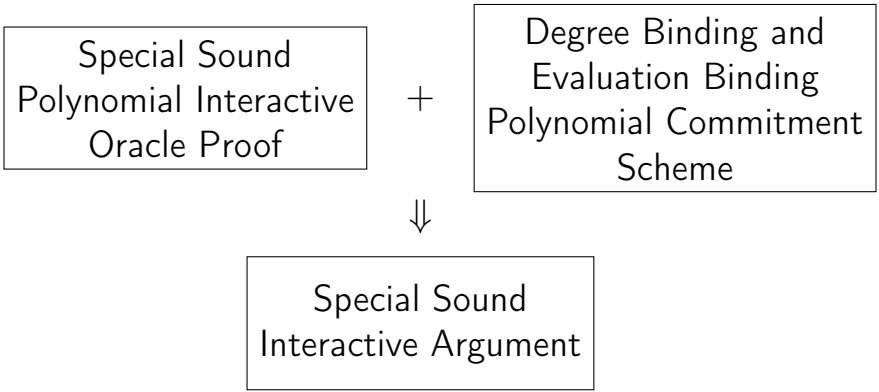
Motivation


Interactive Argument
FS
Non-interactive Argument
Previous work
Knowledge Soundness
Might not have Knowledge Soundness

Motivation

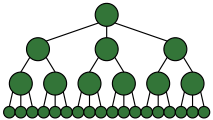

Interactive Argument
FS
Non-interactive Argument
Previous work
Knowledge Soundness
Might not have Knowledge Soundness
Our work
Special Soundness
Knowledge Soundness

Main Theorem



Special Soundness for Polynomial Interactive Oracle Proof

Tree of accepting transcripts



→

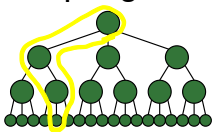
Extractor

→

Witness w

Special Sound Polynomial Interactive Oracle Proof

Tree of accepting transcripts



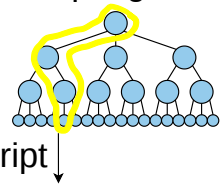
Extractor

Witness w

$$\text{transcript}_{\text{PIOP}} = \left(f_1, \varrho_1, \ldots, f_\mu, \varrho_\mu, f_{\mu+1}, \left(z_i, \bar{f}_{S_i}\right)_{i=1}^{Q}\right) \quad .$$

Special Soundness for Interactive Argument

Tree of accepting transcripts

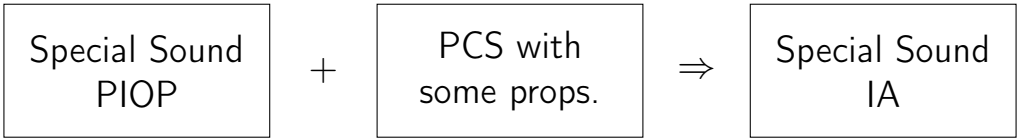


Extractor

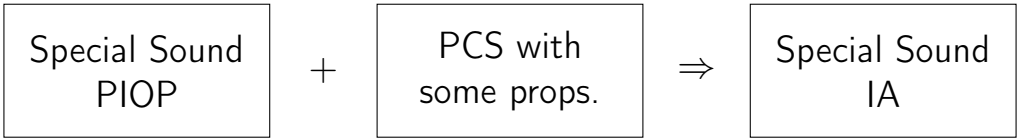
Witness w

$$\text{transcript}_{\mathsf{IA}} = \left(C_1, \varrho_1, \ldots, C_\mu, \varrho_\mu, C_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad .$$

Special Sound Interactive Argument



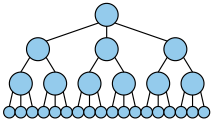
Special Sound Interactive Argument



We need to construct Interactive Argument Extractor  . Note that we have access to the PIOP Extractor  .

Main Theorem Proof Sketch

Tree of accepting transcripts



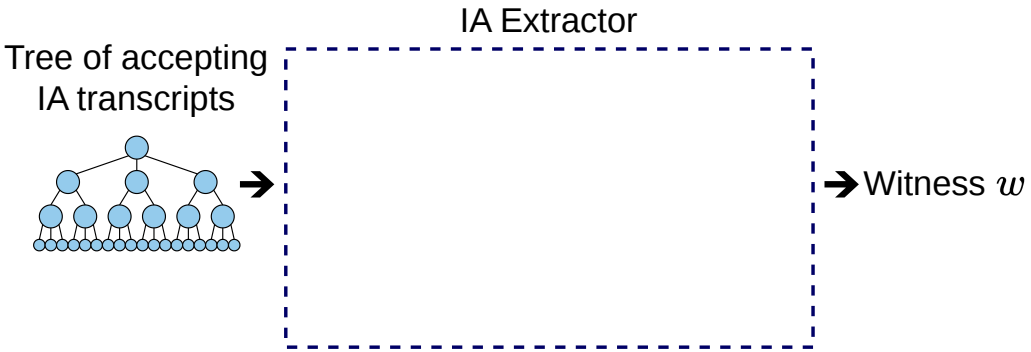
➔

Extractor

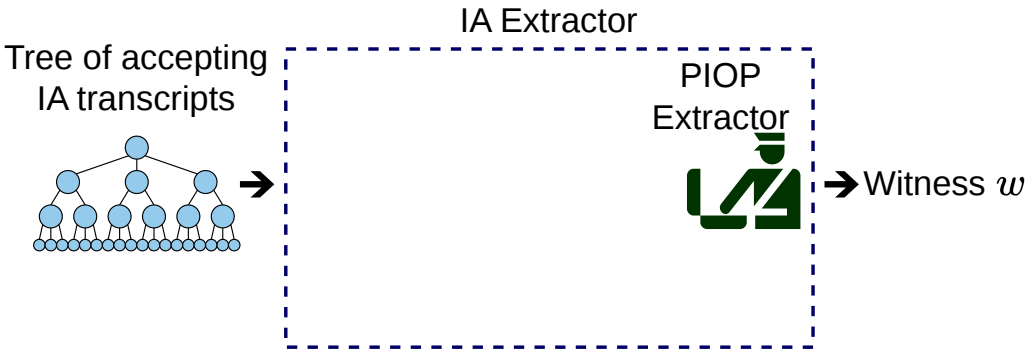
➔

Witness w

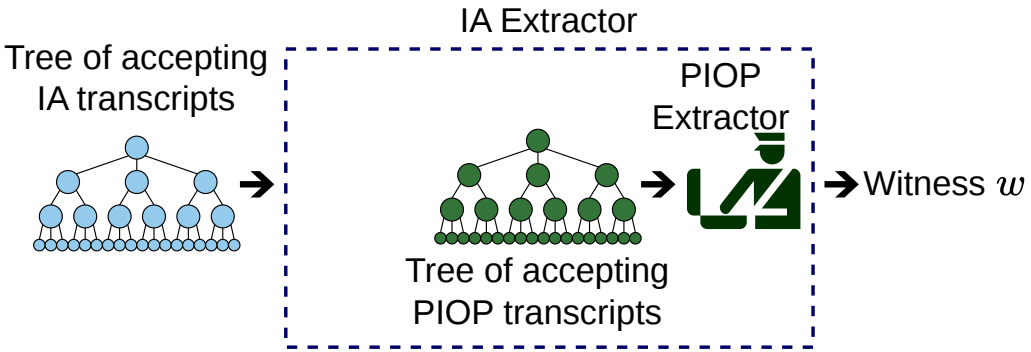
Main Theorem Proof Sketch



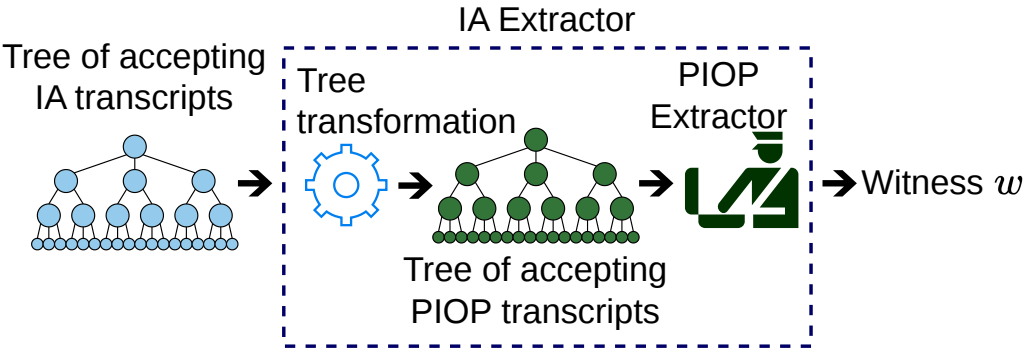
Main Theorem Proof Sketch



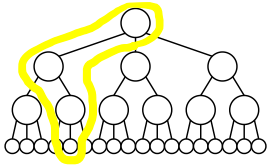
Main Theorem Proof Sketch



Main Theorem Proof Sketch



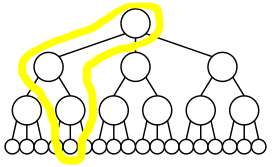
Constructing the Tree Transformation: Transcripts



$$\text{transcript}_{\text{IA}} = \left(C_1, \varrho_1, \ldots, C_\mu, \varrho_\mu, C_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\text{transcript}_{\text{PIOP}} = \left(f_1, \varrho_1, \ldots, f_\mu, \varrho_\mu, f_{\mu+1}, (z_i, \bar{f}_{S_i})_{i=1}^{Q}\right) \quad .$$

Constructing the Tree Transformation: Transcripts

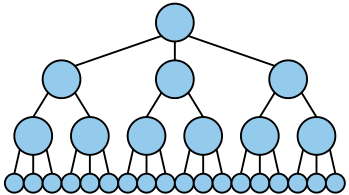


$$\text{transcript}_{\text{IA}} = \left(C_1, \varrho_1, \ldots, C_\mu, \varrho_\mu, C_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\text{transcript}_{\text{PIOP}} = \left(f_1, \varrho_1, \ldots, f_\mu, \varrho_\mu, f_{\mu+1}, (z_i, \bar{f}_{S_i})_{i=1}^{Q}\right) \quad .$$

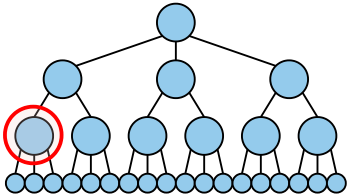
Recovering Polynomials from the Tree of Accepting Transcripts

Fix a commitment C and choose all places in the tree, where the corresponding polynomial is queried.



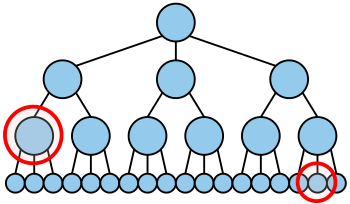
Recovering Polynomials from the Tree of Accepting Transcripts

Fix a commitment C and choose all places in the tree, where the corresponding polynomial is queried.



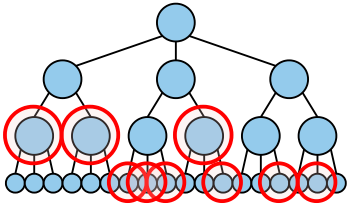
Recovering Polynomials from the Tree of Accepting Transcripts

Fix a commitment C and choose all places in the tree, where the corresponding polynomial is queried.



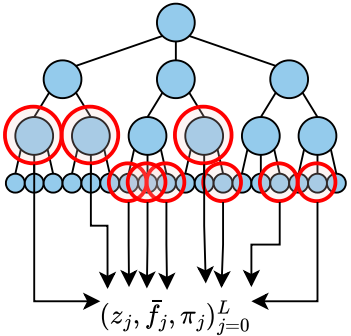
Recovering Polynomials from the Tree of Accepting Transcripts

Fix a commitment C and choose all places in the tree, where the corresponding polynomial is queried.



Recovering Polynomials from the Tree of Accepting Transcripts

Fix a commitment C and choose all places in the tree, where the corresponding polynomial is queried.



Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$.

- What if $z_i = z_j$ but $\bar{f}_i \neq \bar{f}_j$?

Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$.

- What if $z_i = z_j$ but $\bar{f}_i \neq \bar{f}_j$?
- This means the underlying *polynomial* behind the commitment is not a function.

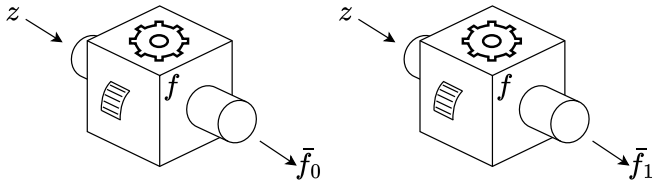
Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$.

- What if $z_i = z_j$ but $\bar{f}_i \neq \bar{f}_j$?
- This means the underlying *polynomial* behind the commitment is not a function.
- Assume it is hard to find such a situation.

Evaluation Binding

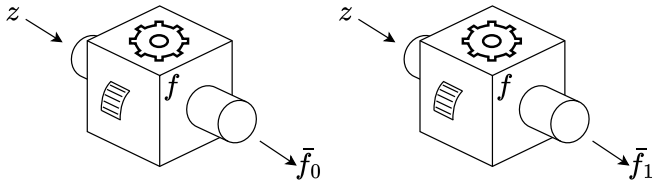
- It should be difficult for a probabilistic polynomial time adversary $\mathcal{A}$ to find $(C, z, \bar{f}_0, \pi_0, \bar{f}_1, \pi_1)$ such that V accepts and $\bar{f}_0 \neq \bar{f}_1$.



$$\bar{f}_0 \neq \bar{f}_1$$

Evaluation Binding

- It should be difficult for a probabilistic polynomial time adversary $\mathcal{A}$ to find $(C, z, \bar{f}_0, \pi_0, \bar{f}_1, \pi_1)$ such that V accepts and $\bar{f}_0 \neq \bar{f}_1$.



- Oracle holds functions that give one output per input.

Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$ and for all $z_i = z_j$ it holds $\bar{f}_i = \bar{f}_j$, thus we can interpolate and get $\tilde{f} \leftarrow \text{Interpolate}((z_j, \bar{f}_j)_{j=0}^{L})$.

- What if $\deg \tilde{f} > n$?

Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$ and for all $z_i = z_j$ it holds $\bar{f}_i = \bar{f}_j$, thus we can interpolate and get $\tilde{f} \leftarrow \text{Interpolate}((z_j, \bar{f}_j)_{j=0}^{L})$.

- What if $\deg \tilde{f} > n$?
- This means that we cannot replace C in the tree with $\tilde{f}$.

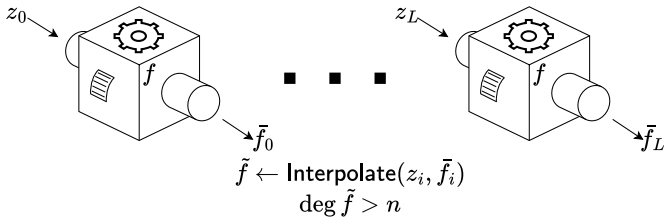
Recovering Polynomials from the Tree of Accepting Transcripts

We have $(z_j, \bar{f}_j, \pi_j)_{j=1}^{L}$ and for all $z_i = z_j$ it holds $\bar{f}_i = \bar{f}_j$, thus we can interpolate and get $\tilde{f} \leftarrow \text{Interpolate}((z_j, \bar{f}_j)_{j=0}^{L})$.

- What if $\deg \tilde{f} > n$?
- This means that we cannot replace C in the tree with $\tilde{f}$.
- Assume it is hard to find such a situation.

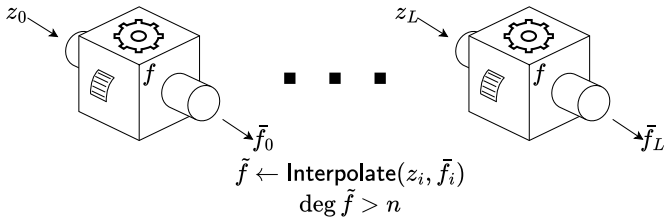
Degree Binding

- It should be difficult for a probabilistic polynomial time adversary $\mathcal{A}$ to find $(C, (z_i, \bar{f}_i, \pi_i)_{i=0}^{L})$ such that V accepts, $\tilde{f} \leftarrow \mathsf{Interpolate}(z_i, \bar{f}_i, \pi_i)_{i=0}^{L})$ and $\deg \tilde{f} > n$.



Degree Binding

- It should be difficult for a probabilistic polynomial time adversary $\mathcal{A}$ to find $(C, (z_i, \bar{f}_i, \pi_i)_{i=0}^{L})$ such that V accepts, $\tilde{f} \leftarrow \mathsf{Interpolate}(z_i, \bar{f}_i, \pi_i)_{i=0}^{L})$ and $\deg \tilde{f} > n$.



- Oracle holds polynomials with degree up to n.

Constructing the Tree Transformation

$$\text{transcript}_{\text{IA}} = \left(C_1, \varrho_1, \ldots, C_\mu, \varrho_\mu, C_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\text{transcript} = \left(\tilde{f}_1, \varrho_1, \ldots, \tilde{f}_\mu, \varrho_\mu, \tilde{f}_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\text{transcript}_{\text{PIOP}} = \left(f_1, \varrho_1, \ldots, f_\mu, \varrho_\mu, f_{\mu+1}, (z_i, \bar{f}_{S_i})_{i=1}^{Q}\right) \quad .$$

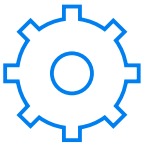
Constructing the Tree Transformation

$$\mathsf{transcript}_{\mathsf{IA}} = \left(C_1, \varrho_1, \ldots, C_\mu, \varrho_\mu, C_{\mu+1}, (z_i, \bar{f}_{S_i}, \pi_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\mathsf{transcript} = \left(\tilde{f}_1, \varrho_1, \ldots, \tilde{f}_\mu, \varrho_\mu, \tilde{f}_{\mu+1}, (z_i, \bar{f}_{S_i})_{i=1}^{Q}\right) \quad ,$$

$$\mathsf{transcript}_{\mathsf{PIOP}} = \left(f_1, \varrho_1, \ldots, f_\mu, \varrho_\mu, f_{\mu+1}, (z_i, \bar{f}_{S_i})_{i=1}^{Q}\right) \quad .$$

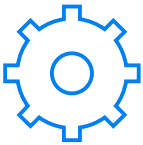
Constructing the Tree Transformation



In conclusion, we do two main operations in the tree transformation:

1. Replace commitments with interpolated polynomials. $C \to \tilde{f}$

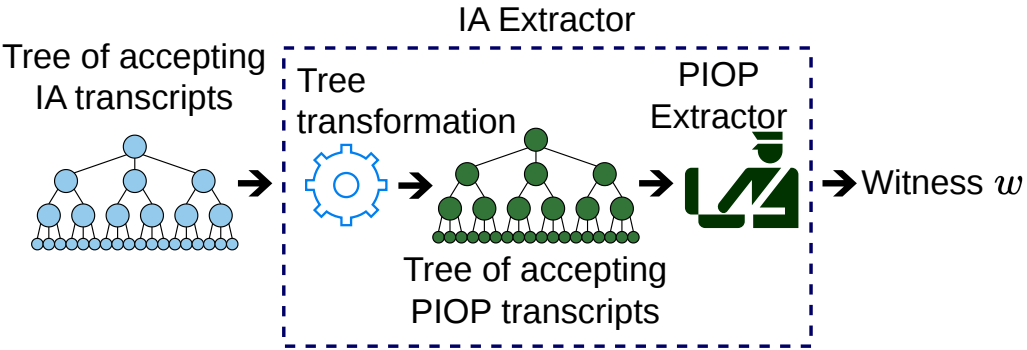
Constructing the Tree Transformation



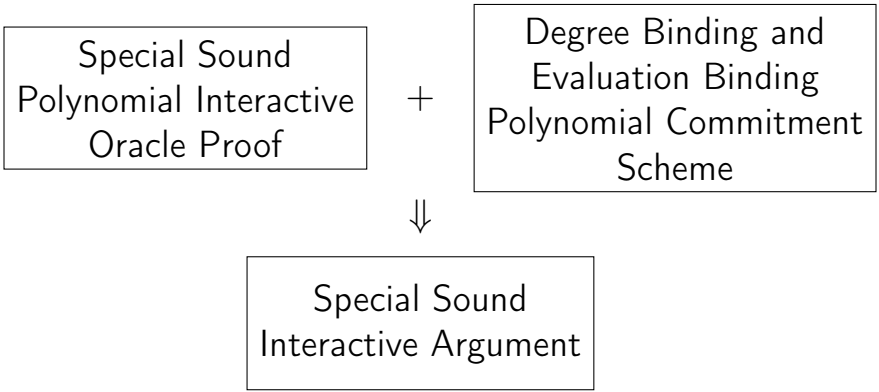
In conclusion, we do two main operations in the tree transformation:

1. Replace commitments with interpolated polynomials. $C \to \tilde{f}$
2. Delete proofs. π_j

Main Theorem Proof Sketch



Main Theorem



Other Results from the Paper

We also considered the following

- What if the Polynomial Commitment Scheme had an interactive opening and verification phase?

Other Results from the Paper

We also considered the following

- What if the Polynomial Commitment Scheme had an interactive opening and verification phase?
- There are other binding properties under which the theorem also works. Interpolation Binding + Binding.

Other Results from the Paper

We also considered the following

- What if the Polynomial Commitment Scheme had an interactive opening and verification phase?
- There are other binding properties under which the theorem also works. Interpolation Binding + Binding.
- What are the relationships between all these binding notions?

Other Results from the Paper

We also considered the following

- What if the Polynomial Commitment Scheme had an interactive opening and verification phase?
- There are other binding properties under which the theorem also works. Interpolation Binding + Binding.
- What are the relationships between all these binding notions?
- Binding properties of KZG Polynomial Commitment Scheme and its optimizations.

See [KLPS26] for the details.

Conclusion

- We expanded the theoretical background behind Polynomial Interactive Oracle Proofs and Polynomial Commitment Schemes, and provided a new way of proving tight knowledge soundness of SNARKs.

Conclusion

- We expanded the theoretical background behind Polynomial Interactive Oracle Proofs and Polynomial Commitment Schemes, and provided a new way of proving tight knowledge soundness of SNARKs.
- Future questions:
  - What about Polynomial Commitment Schemes that have an interactive commitment phase?

Conclusion

- We expanded the theoretical background behind Polynomial Interactive Oracle Proofs and Polynomial Commitment Schemes, and provided a new way of proving tight knowledge soundness of SNARKs.
- Future questions:
  - What about Polynomial Commitment Schemes that have an interactive commitment phase?
  - Do other Polynomial Commitment Schemes have Degree Binding and under which assumptions?

Conclusion

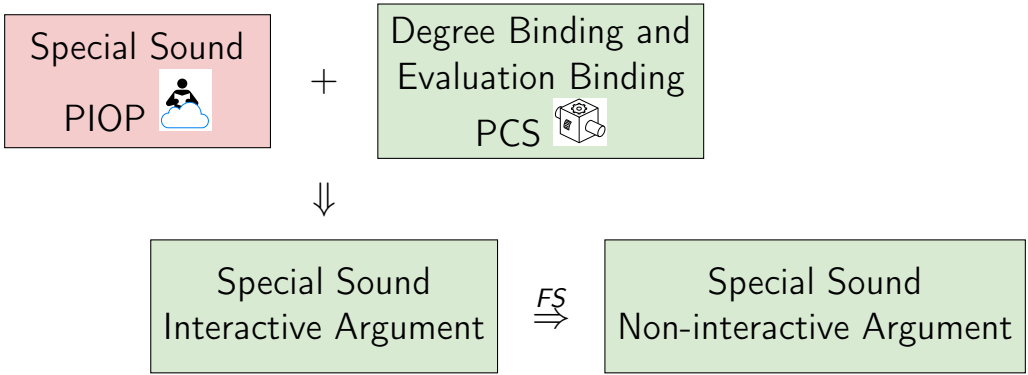
- We expanded the theoretical background behind Polynomial Interactive Oracle Proofs and Polynomial Commitment Schemes, and provided a new way of proving tight knowledge soundness of SNARKs.
- Future questions:
  - What about Polynomial Commitment Schemes that have an interactive commitment phase?
  - Do other Polynomial Commitment Schemes have Degree Binding and under which assumptions?
  - Can we do something similar with multilinear polynomials?

Conclusion

- We expanded the theoretical background behind Polynomial Interactive Oracle Proofs and Polynomial Commitment Schemes, and provided a new way of proving tight knowledge soundness of SNARKs.
- Future questions:
  - What about Polynomial Commitment Schemes that have an interactive commitment phase?
  - Do other Polynomial Commitment Schemes have Degree Binding and under which assumptions?
  - Can we do something similar with multilinear polynomials?
  - Can we make a similar theory for stronger security notions like simulation extractability and universal composability?

Thank you for listening!

Questions


Special Sound
PIOP
+
Degree Binding and
Evaluation Binding
PCS
⇓
Special Sound
Interactive Argument
FS
⇒
Special Sound
Non-interactive Argument

References

[KLPS26] Erki Külaots, Helger Lipmaa, Roberto Parisella, and Janno Siim.

Special soundness and binding properties: A framework for tightly secure zk-SNARKs.

Cryptology ePrint Archive, Paper 2026/326, 2026.

URL: https://eprint.iacr.org/2026/326.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg.

Constant-size commitments to polynomials and their applications.

In International conference on the theory and application of cryptology and information security, pages 177–194. Springer, 2010.