How to Compile Polynomial IOP into Simulation-Extractable SNARKs: A Modular Approach

Markulf Kohlweiss, Mahak Pancholi, Akira Takahashi Eprint: 2023/1067

From Polynomial IOP and Commitments to Non-malleable zkSNARKs

Antonio Faonio, Dario Fiore, Markulf Kohlweiss, Luigi Russo, Michal Zajac

Eprint: 2023/569

This Talk

- **Non-Malleability** of zkSNARKs **compiled** from **Polynomial Interactive Oracle Proofs (PIOP)**.

This Talk

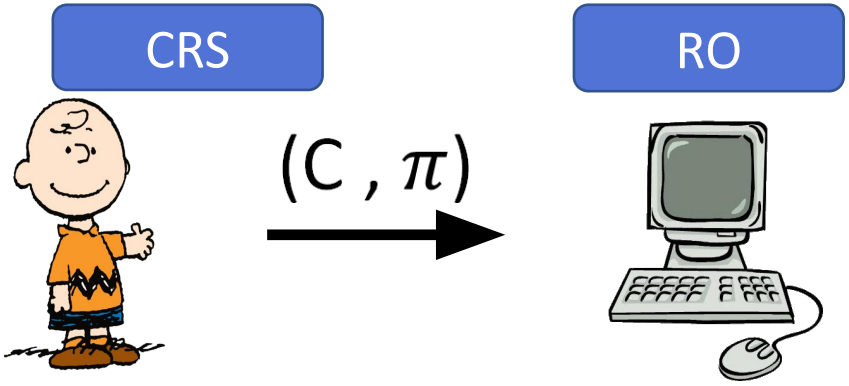
- **Non-Malleability** of zkSNARKs **compiled** from **Polynomial Interactive Oracle Proofs (PIOP)**.
- Identify easy to verify properties from **Polynomial Commitments** and **PIOP**.

This Talk

- **Non-Malleability** of zkSNARKs **compiled** from **Polynomial Interactive Oracle Proofs (PIOP)**.
- Identify easy to verify properties from **Polynomial Commitments** and **PIOP**.
- Implications: (variants of) PLONK, MARLIN, LUNAR + KZG are non-malleable.

Why Non-Malleability?

Claim: I know w s.t. C(w)=1.

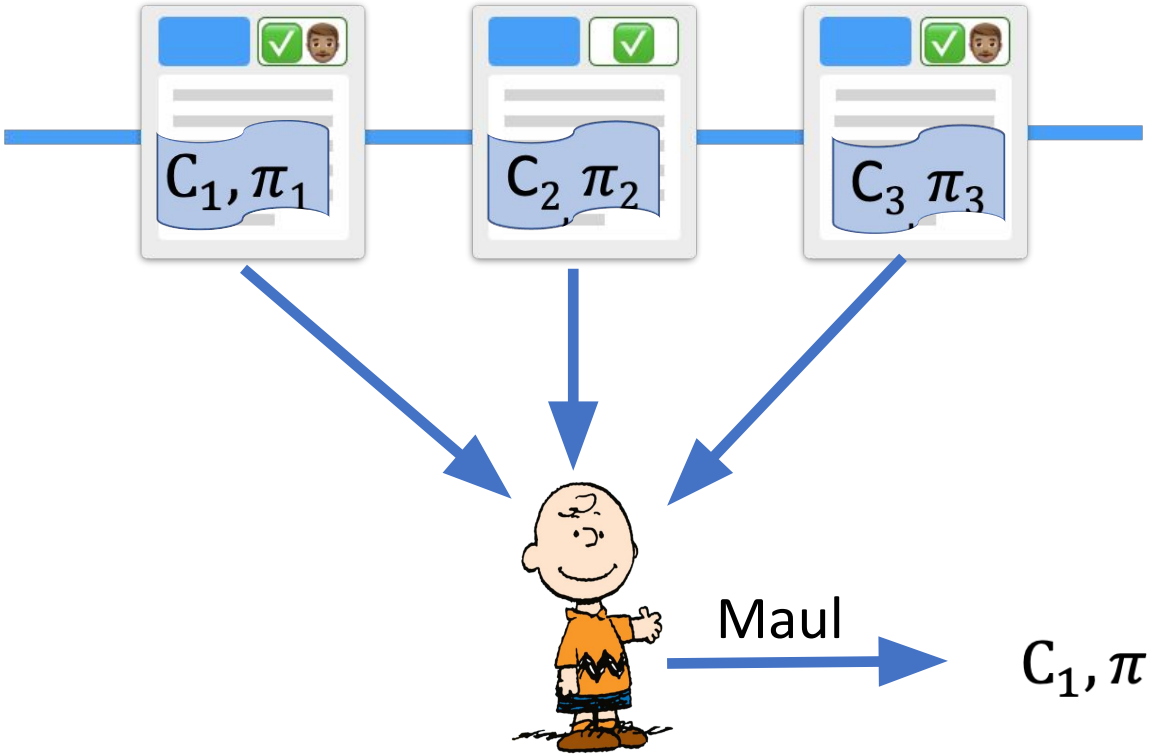


Input: C,w
Output: π

Input: (C , π)
Output: w* s.t.
If π verifies then C(w*)=1.

- Succinct
- Non-interactive
- Complete
- Zero-Knowledge
- Knowledge Soundness (KS)

Malleability Attack

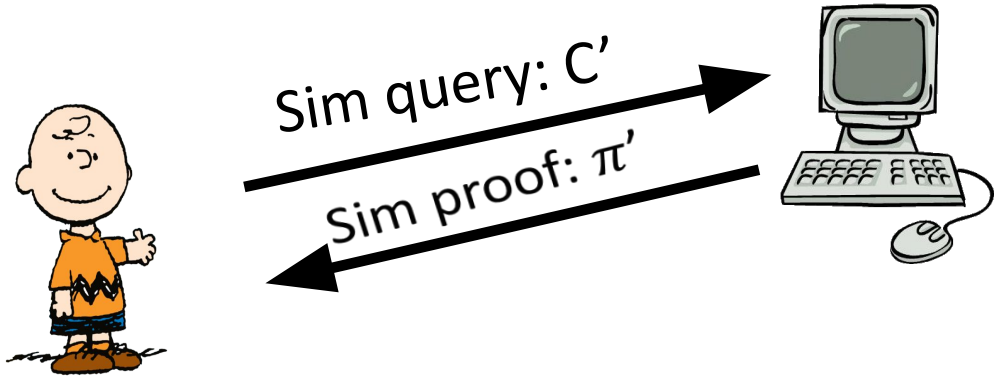


Without knowing the witness for C_1 !

Simulation-Extractability (Sim-Ext)

Claim: I know w s.t. C(w)=1.

CRS | RO



- Prover sees other proofs: Sim queries.
- Extractor still succeeds.

Simulation-Extractability (Sim-Ext)

Claim: I know w s.t. C(w)=1.

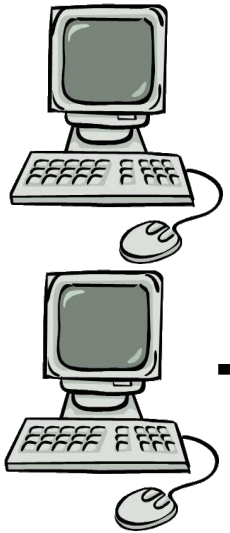
CRS | RO

- Prover sees other proofs: Sim queries.
- Extractor still succeeds.

Sim query: C'

Sim proof: π'

(C, π)



w* s.t.

- if C was not a sim query and
- if π verifies,

C(w*) = 1.

Gaps in Previous Results

[FKMV12,GKK^{+}22,GOP^{+}23,DG23]

- Analysis is protocol specific
- GKK^{+}21: Plonk, Marlin, Sonic
- GOP^{+}23: Bulletproofs
- DG23: Bulletproofs, Spartan

Gaps in Previous Results

[FKMV12,GKK+22,GOP+23,DG23]

- Analysis is protocol specific
- GKK+21: Plonk, Marlin, Sonic
- GOP+23: Bulletproofs
- DG23: Bulletproofs, Spartan

Question: Is it possible to prove sim-ext for a large class of existing zkSNARKs without analysing each zkSNARK separately?

Question: Is it possible to prove sim-ext for a large class of existing zkSNARKs without analysing each zkSNARK separately?

Polynomial Interactive Oracle Proof

Polynomial Commitment Scheme

Fiat-Shamir

Question: Is it possible to prove sim-ext for a large class of existing zkSNARKs without analysing each zkSNARK separately?

Polynomial Interactive Oracle Proof

Polynomial Commitment Scheme

Fiat-Shamir

PIOP: Marlin [CHM+20], Lunar [CFF+21], Plonk [GWC19], Sonic [MBKM19],....

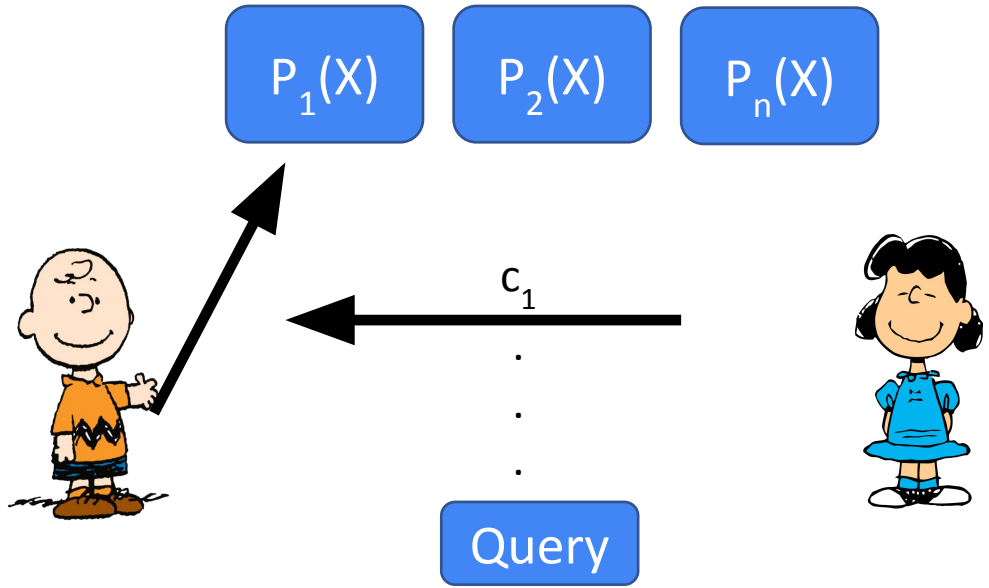
PCOM: KZG [KZG10]

PIOP + KS PCOM + FS = KS zkSNARK

Polynomial Interactive Oracle Proof

Polynomial Commitment Scheme

Fiat-Shamir



PIOP + KS PCOM + FS = KS zkSNARK

Polynomial Interactive Oracle Proof

Polynomial Commitment Scheme

Fiat-Shamir

$P_1(X)$ $P_2(X)$ $P_n(X)$

a

$c_1, c_2 \ldots$

Query

$(P_1, a), (P_2, b) \ldots$

$P_1(a), P_2(b), \ldots, P_n(z)$

Output: Decision bit

PIOP + KS PCOM + FS = KS zkSNARK

Polynomial Interactive Oracle Proof

Polynomial Commitment Scheme

Fiat-Shamir

Com(P_1)

c_1

…

Query

(P_1,a),(P_2,b)…

(y_1, π_1), …

RO

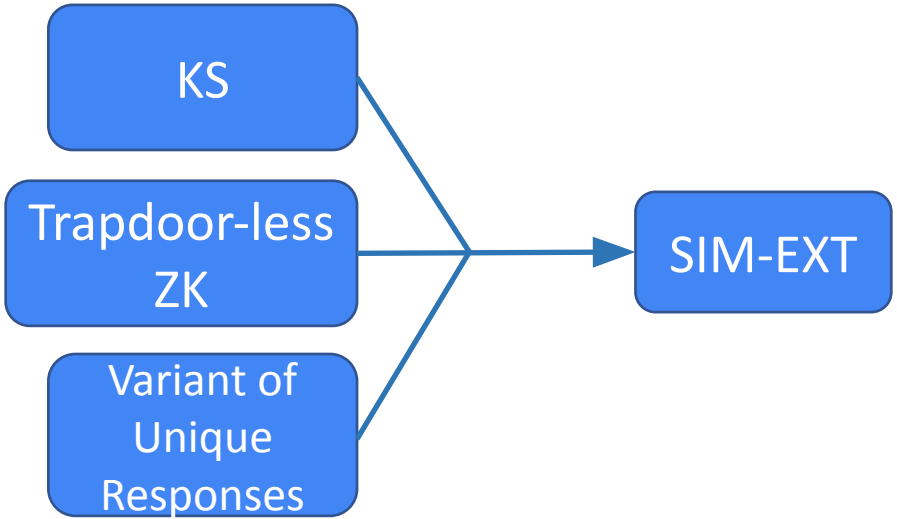
Polynomial commitment:

com ← Commit(P)

π_e ← Eval(P,z,y): π_e guarantees P(z) = y.

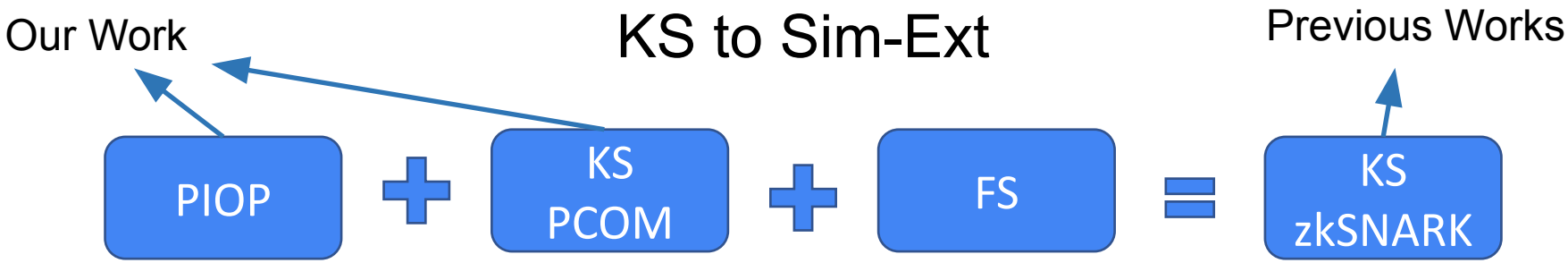
Proof: (Com_1, c_1, Com_2,…, y1, π_1,…)

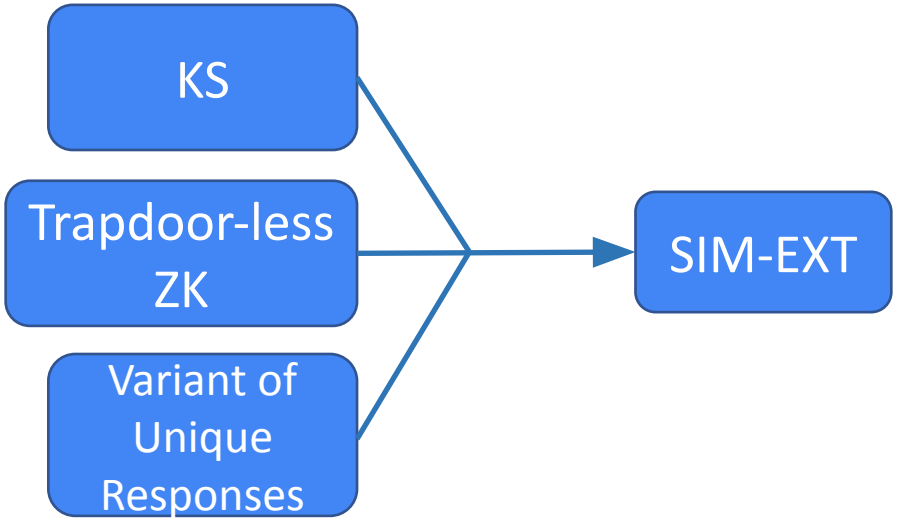
KS to Sim-Ext



- Proof Template from previous papers [FKMV12,GKK+22,GOP+23,DG23]

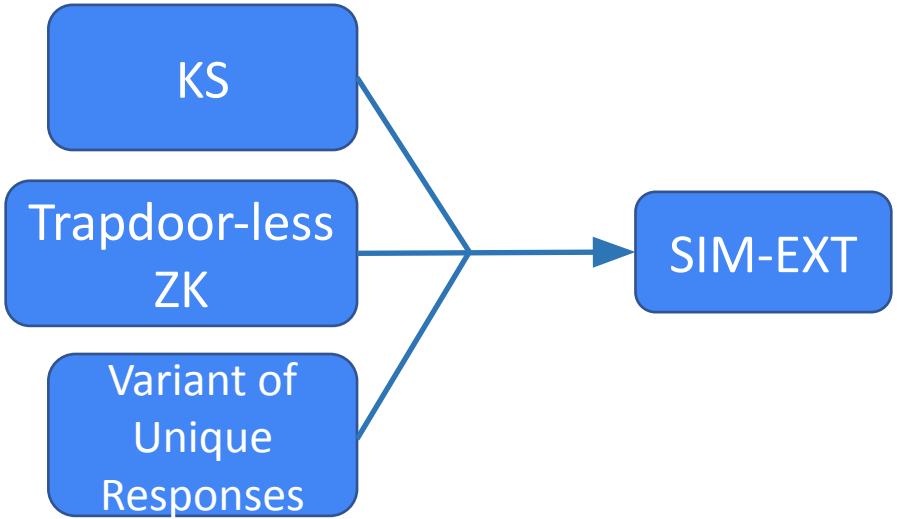
KS to Sim-Ext





- Proof Template from previous papers [FKMV12,GKK+22,GOP+23,DG23]
- But now, applied at the compiler level

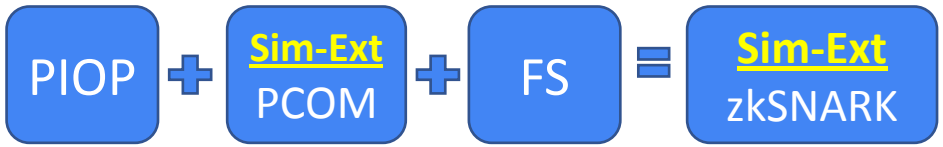
KS to Sim-Ext



- Proof Template from previous papers [FKMV12,GKK+22,GOP+23,DG23]
- But now, applied at the compiler level
- Weak Unique Response (WUR)

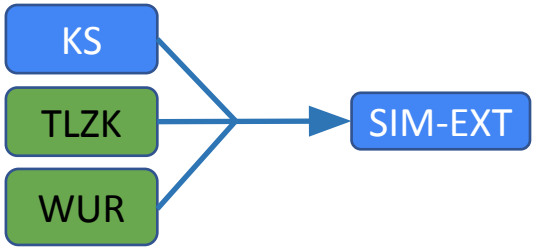
KS to Sim-Ext

- More direct approach
- New definition for Sim-Ext for PCOM



Mahak’s approach

Mahak's Approach

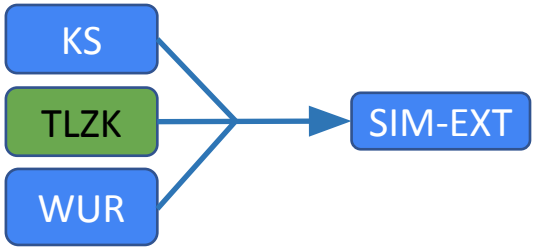


	PIOP	PCOM
TLZK		
WUR		

Identify Properties
From PIOP and PCOM

Mahak's Approach

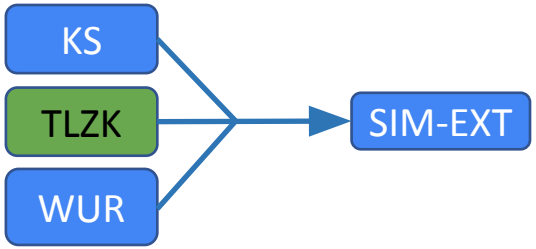
ZK Simulator does not use a trapdoor for simulation.



	PIOP	PCOM
TLZK	HVZK	Hiding
TLZK		

Standard properties

Mahak's Approach

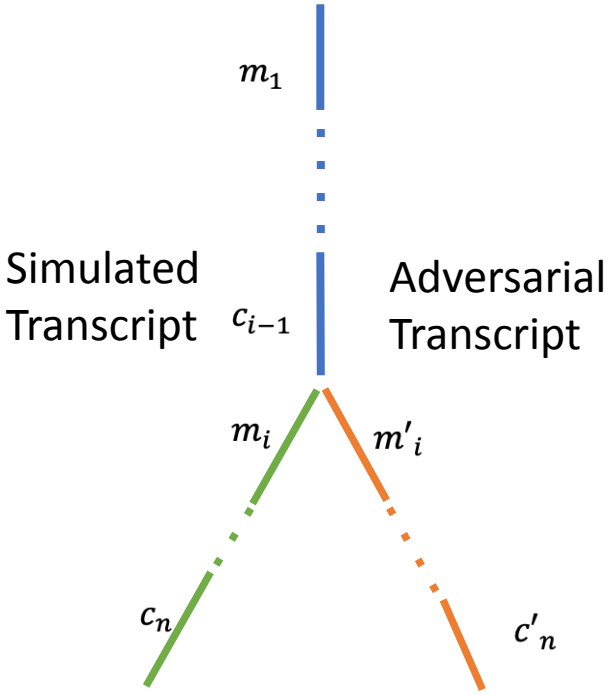


New property

	PIOP	PCOM
TLZK	HVZK	Hiding
TLZK	**Ψ-HVZK**	Weak Hiding

(check eprint)

Mahak's Approach



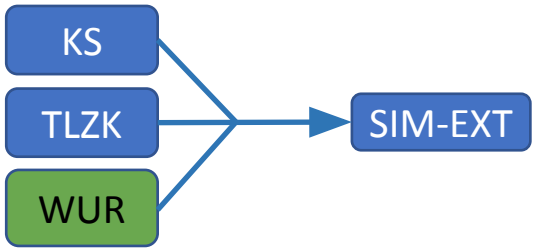
KS

Trapdoor-less ZK

Weak Unique Responses

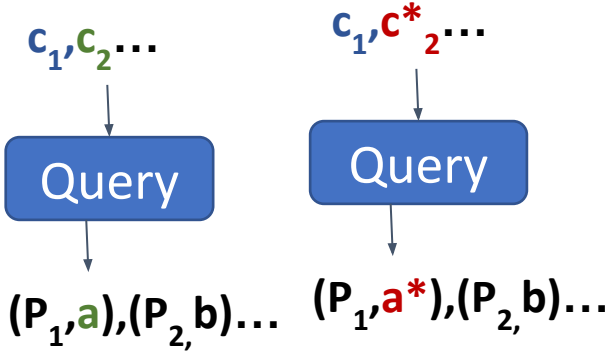
SIM-EXT

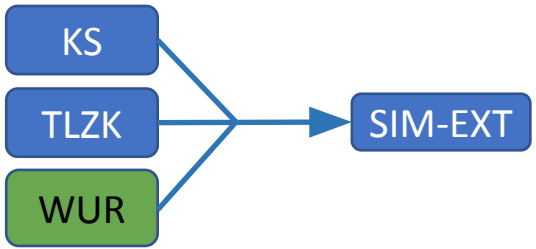
Mahak's Approach



	PIOP	PCOM
WUR	High min-entropy for Query,	
WUR		

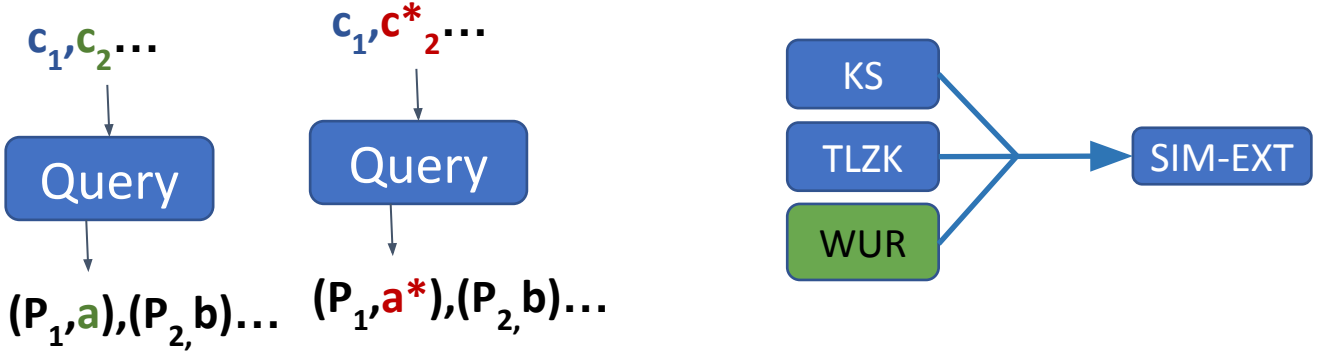
Mahak's Approach





	PIOP	PCOM
WUR	High min-entropy for Query,	
WUR		

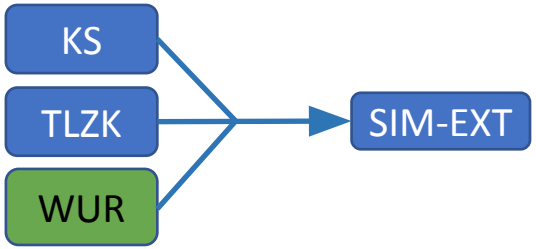
Mahak's Approach



Easy to Verify

	PIOP	PCOM
WUR	High min-entropy for Query, P_1 is queried on < deg +1 points	
WUR		

Mahak's Approach

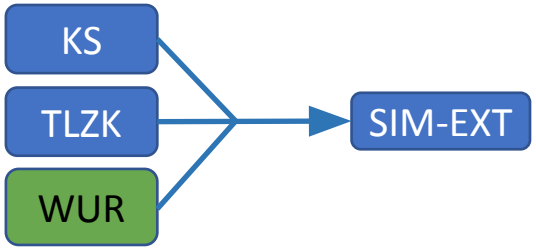


(mostly) Standard

Easy to Verify

	PIOP	PCOM
WUR	High min-entropy for Query, P_1 is queried on < deg +1 points	Hiding, Eval Binding, Unique Proofs
WUR		

Mahak's Approach

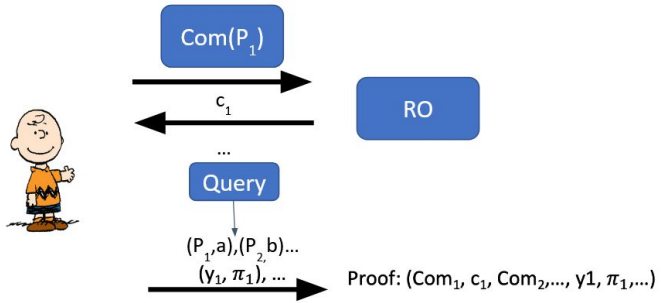


New property

	PIOP	PCOM
WUR	High min-entropy for Query, P_1 is queried on < deg +1 points	Hiding, Eval Binding, Unique Proofs
WUR	High min-entropy for Query, **Ψ-NEXP**	**Weak** Hiding, Eval Binding, Unique Proofs

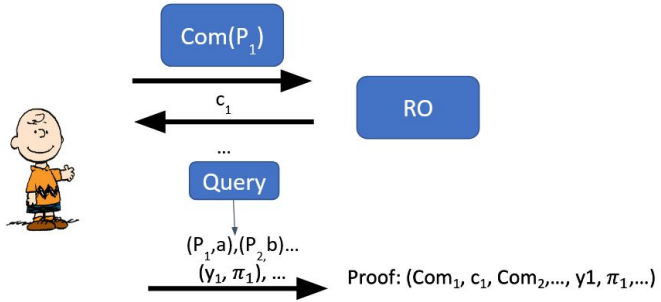
Luigi’s approach

PIOP+PCom+FS



zkSNARKs compiled in this way are not (always) simulation-extractable

PIOP+PCom+FS



zkSNARKs compiled in this way are not (always) simulation-extractable

If PCOM is (fully) malleable, one can transform a simulated proof on a bunch of commitments into another proof

Luigi's approach

Luigi's approach

KZG?

This may work, but **would it capture existing constructions**?

KZG Polynomial Commitment

$$\text{srs} \leftarrow \left(\left[1,\, s, \ldots, s^d\right]_1,\, [1, s]_2 \right)$$

$$\text{Com}(p) \rightarrow [p(s)]_1$$

$$\text{Open}(p, x, y) \rightarrow \left[\frac{p(s) - p(x)}{s - x} \right]_1$$

$$\text{Verify}(C, x, y, \pi) \rightarrow 1 \iff e(C - [y]_1,\, [1]_2) = e([\pi]_1, [s - x]_2)$$

Malleability of KZG

Given a proof π for (C,x,y), it is a valid proof also on:

- Same point x: $(C + [\delta]_1, x, y+\delta)$

Malleability of KZG

Given a proof π for (C,x,y), it is a valid proof also on:

- Same point x: $(C + [\delta]_1, x, y+\delta)$
- Arbitrary point x*: $(C - (x^*-x)\pi, x^*, y)$

Malleability of KZG

Given a proof π for (C,x,y), it is a valid proof also on:

- Same point x: $(C + [\delta]_1, x, y+\delta)$
- Arbitrary point x*: $(C - (x^*-x)\pi, x^\wedge*, y)$

These attacks are not enough to break SE of zkSNARKs!

If the PIOP is **compiler-safe**: polynomials are evaluated on last-round challenge

Luigi's approach (2)

We introduce **policy-based simulation-extractability**, a generalization of sim-ext that can capture relaxed variants

Luigi's approach (2)

We introduce **policy-based simulation-extractability**, a generalization of sim-ext that can capture relaxed variants

We identify a policy that is:

- sufficient for the compiler to work
- met by KZG

Policy-based Simulation-extractability

If policy is satisfied:

Claim: I know w s.t. C(w)=1

CRS

RO

(C, π)

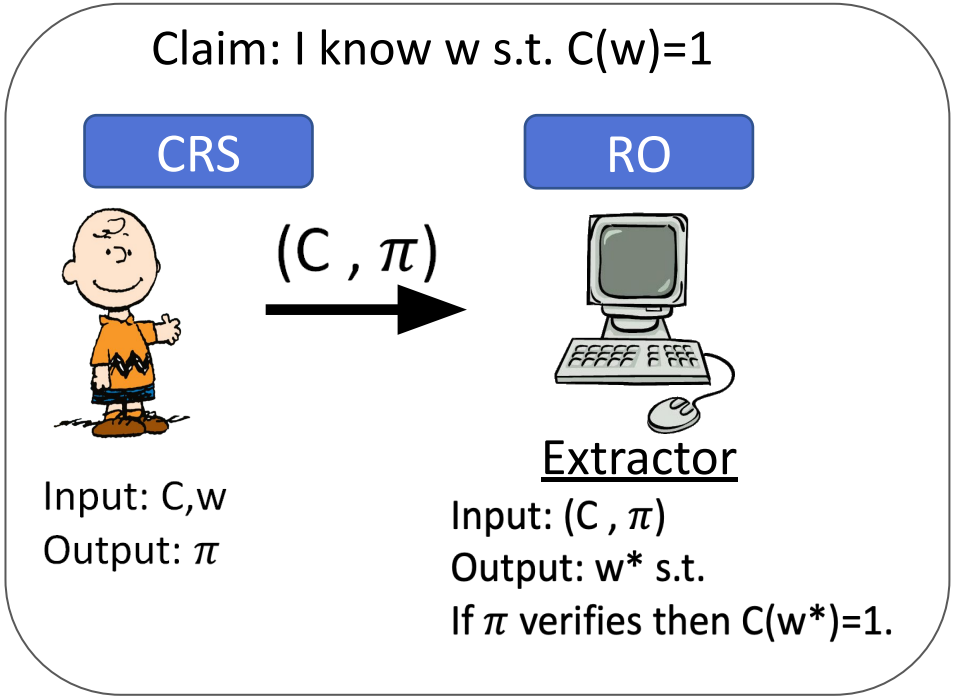
Extractor

Input: C,w
Output: π

Input: (C, π)
Output: w* s.t.
If π verifies then C(w*)=1.

Policy-based Simulation-extractability

If policy is satisfied:

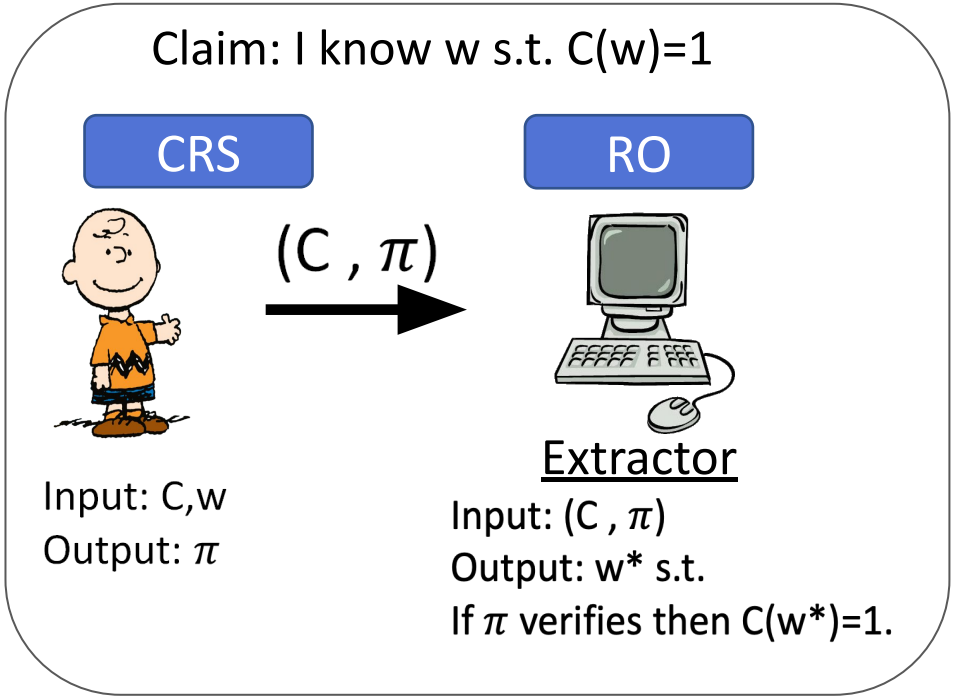


- **Weak sim-ext**:

  the statement must be fresh

Policy-based Simulation-extractability

If policy is satisfied:



- **Weak sim-ext**:

  the statement must be fresh

- **True sim-ext**:

  can only ask true simulations

A policy for KZG (in AGM+RO)

Simulation policy

- queries on points x_i non adaptively chosen
- algebraic consistency: queries admit solutions

  e.g. cannot ask (C,x,y) and (C,x, y')

Extraction policy

- forgery on a (sufficiently) random point x*, chosen after C*

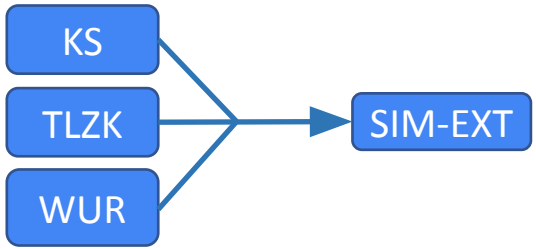
Luigi's approach (3)

- Non adaptive algebraic verifier: queries are "independent" of the statement
- Commitment simulator: there is a simulation strategy that is linear w.r.t. a vector of simulated commitments for free if commitment is hiding

To wrap-up

Our Two Approaches: Main differences

Mahak



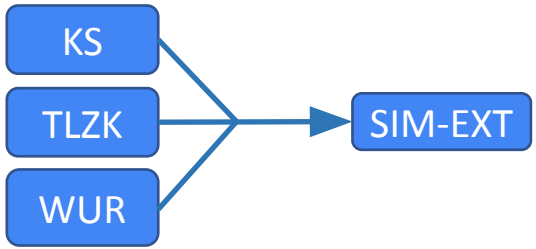
- Trapdoorless Simulation

Luigi

- Trapdoored Simulation

Our Two Approaches: Main differences

Mahak



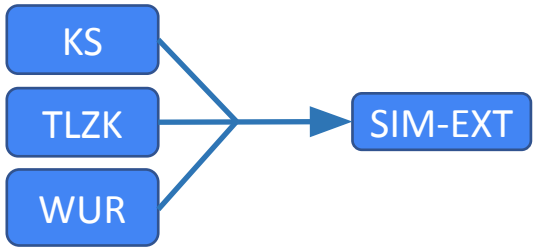
- Trapdoorless Simulation
- WUR from Unique Proofs of Pcom

Luigi

- Trapdoored Simulation
- Sim-ext for Pcom

Our Two Approaches: Main differences

Mahak



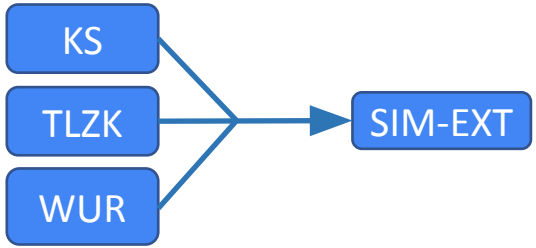
- Trapdoorless Simulation
- WUR from Unique Proofs of Pcom
- Easy to prove for KZG, no AGM

Luigi

- Trapdoored Simulation
- Sim-ext for Pcom
- Somewhat involved for KZG, in AGM

Our Two Approaches: Main differences

Mahak

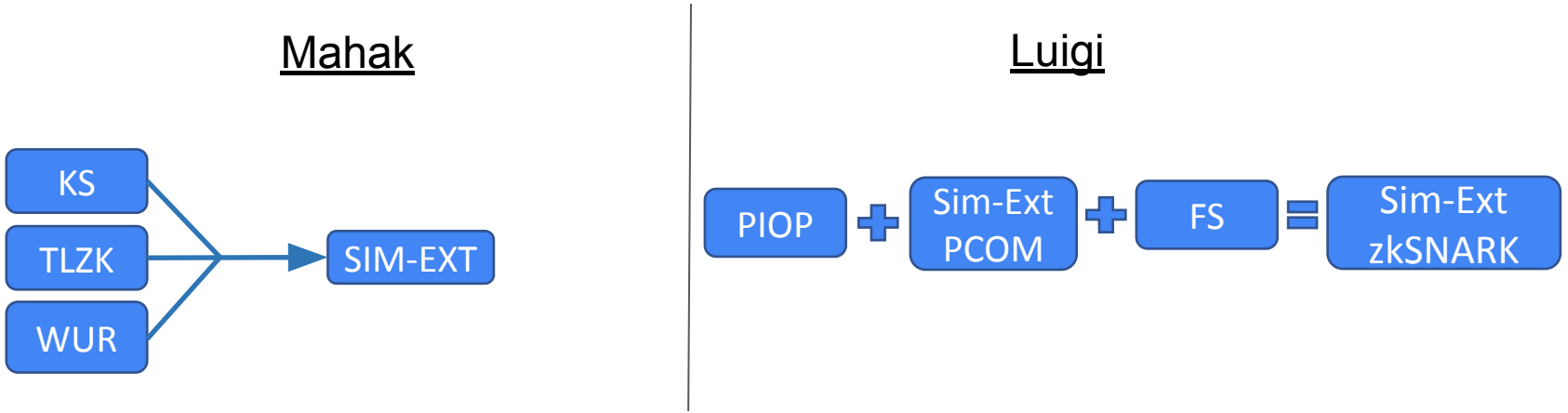


- Trapdoorless Simulation
- WUR from Unique Proofs of Pcom
- Easy to prove for KZG, no AGM
- Plonk with hiding KZG

Luigi

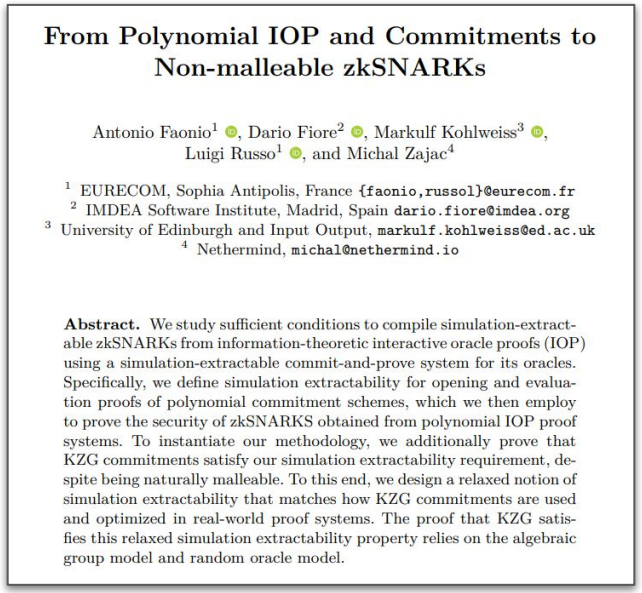
- Trapdoored Simulation
- Sim-ext for Pcom
- Somewhat involved for KZG, in AGM
- Plonk

Our Two Approaches: Common limits



- Slight variants of Marlin and Lunar
- RO programmability
- No linearization trick

Thanks!



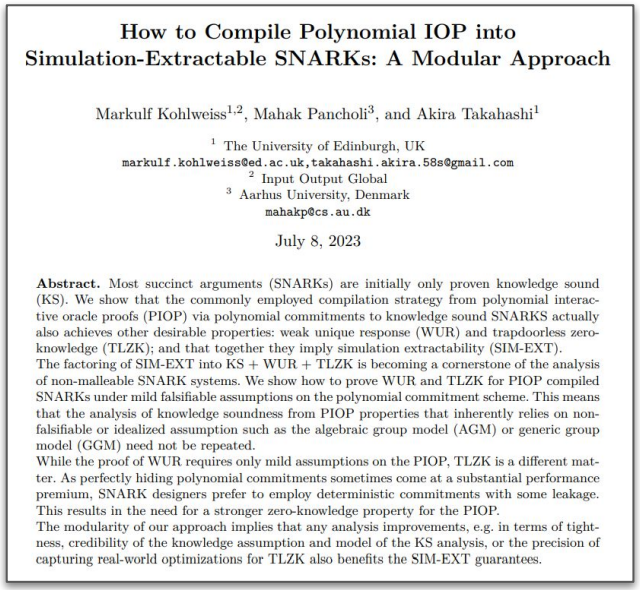
From Polynomial IOP and Commitments to Non-malleable zkSNARKs

Antonio Faonio[1], Dario Fiore[2], Markulf Kohlweiss[3], Luigi Russo[1], and Michal Zajac[4]

[1] EURECOM, Sophia Antipolis, France {faonio,russol}@eurecom.fr
[2] IMDEA Software Institute, Madrid, Spain dario.fiore@imdea.org
[3] University of Edinburgh and Input Output, markulf.kohlweiss@ed.ac.uk
[4] Nethermind, michal@nethermind.io

Abstract. We study sufficient conditions to compile simulation-extractable zkSNARKs from information-theoretic interactive oracle proofs (IOP) using a simulation-extractable commit-and-prove system for its oracles. Specifically, we define simulation extractability for opening and evaluation proofs of polynomial commitment schemes, which we then employ to prove the security of zkSNARKS obtained from polynomial IOP proof systems. To instantiate our methodology, we additionally prove that KZG commitments satisfy our simulation extractability requirement, despite being naturally malleable. To this end, we design a relaxed notion of simulation extractability that matches how KZG commitments are used and optimized in real-world proof systems. The proof that KZG satisfies this relaxed simulation extractability property relies on the algebraic group model and random oracle model.

ia.cr/2023/569



How to Compile Polynomial IOP into Simulation-Extractable SNARKs: A Modular Approach

Markulf Kohlweiss[1,2], Mahak Pancholi[3], and Akira Takahashi[1]

[1] The University of Edinburgh, UK
markulf.kohlweiss@ed.ac.uk,takahashi.akira.58s@gmail.com
[2] Input Output Global
[3] Aarhus University, Denmark
mahakp@cs.au.dk

July 8, 2023

Abstract. Most succinct arguments (SNARKs) are initially only proven knowledge sound (KS). We show that the commonly employed compilation strategy from polynomial interactive oracle proofs (PIOP) via polynomial commitments to knowledge sound SNARKS actually also achieves other desirable properties: weak unique response (WUR) and trapdoorless zero-knowledge (TLZK); and that together they imply simulation extractability (SIM-EXT).
The factoring of SIM-EXT into KS + WUR + TLZK is becoming a cornerstone of the analysis of non-malleable SNARK systems. We show how to prove WUR and TLZK for PIOP compiled SNARKs under mild falsifiable assumptions on the polynomial commitment scheme. This means that the analysis of knowledge soundness from PIOP properties that inherently relies on non-falsifiable or idealized assumption such as the algebraic group model (AGM) or generic group model (GGM) need not be repeated.
While the proof of WUR requires only mild assumptions on the PIOP, TLZK is a different matter. As perfectly hiding polynomial commitments sometimes come at a substantial performance premium, SNARK designers prefer to employ deterministic commitments with some leakage. This results in the need for a stronger zero-knowledge property for the PIOP.
The modularity of our approach implies that any analysis improvements, e.g. in terms of tightness, credibility of the knowledge assumption and model of the KS analysis, or the precision of capturing real-world optimizations for TLZK also benefits the SIM-EXT guarantees.

ia.cr/2023/1067