Plonk is sound and your money is safe

Helger Lipmaa, University of Tartu

Roberto Parisella, Simula UiB

Janno Siim, University of Tartu

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

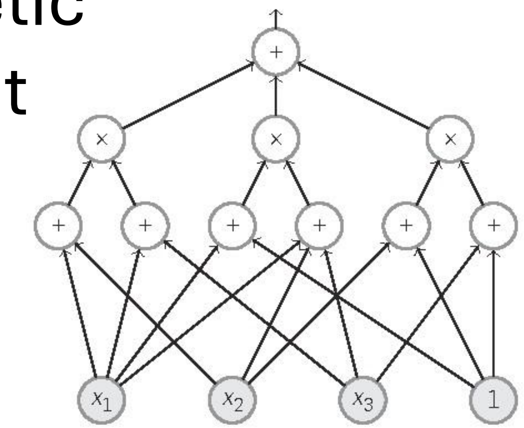
Applications

- Other cryptographic primitives
- Blockchains
- Digital currencies (Zcash)
- Electronic voting systems
- Secure and anonymous authentications
- Outsourced verifiable computation
- And many more …

Circuit satisfiability and constraint systems

Statement x
Public

Arithmetic circuit



Witness w
Private

Statement x
Public

Constraint system

$$F_1(x, w) = G_1(x)$$
$$F_2(x, w) = G_2(x)$$
$$\vdots$$
$$F_k(x, w) = G_k(x)$$

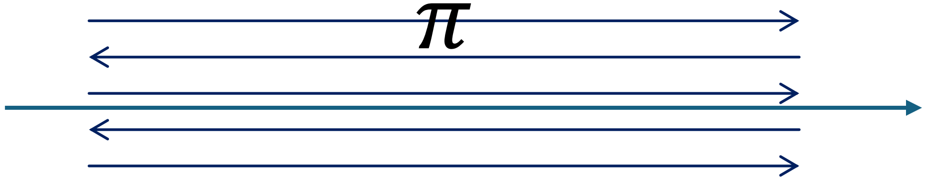
Witness w
Private

Zero-Knowledge Argument / NIZK: Non-Interactive Zero-Knowledge In the SRS Model

Trusted third party: $setup\left(1^{\lambda}\right) \rightarrow srs$

Prover (srs, x, w)

Verifier (srs, x)



π

Accept or reject

Security Properties

Prover (srs, x, w) / Prover (srs, x, aux)

Verifier (srs, x)

Accept

$$F_1(x, w) = G_1(x)$$
$$F_2(x, w) = G_2(x)$$
$$\vdots$$
$$F_k(x, w) = G_k(x)$$

Extractor

knows nothing about the witness

$\rightarrow w$

Security Through Reductions

- Hardness assumption

It is impossible to find a 3-colorability of a graph G

(in reasonable time)

Typical theorem in zero-knowledge:

3-color hard → Our new super-cool scheme enjoyes Knowledge-soundness

Assumption

Security properties

Security Through Reductions

- Hardness assumption

It is impossible to find a 3-colorability of a graph G

(in reasonable time)

Typical proof in zero-knowledge:

Assume efficient $TM\ A$ breaks knowledge-soundness

Design efficient $TM\ B$ on input G:

Simulate valid inputs for A

Call A

Use A's output to find alleged f

If A

breaks knowledge-soundness

Then

f is a 3-colourabilitylity for G

Computational Assumptions

Adversary A

Challenger C

Input (a random big graph G)

Output (a 3-colourability f)

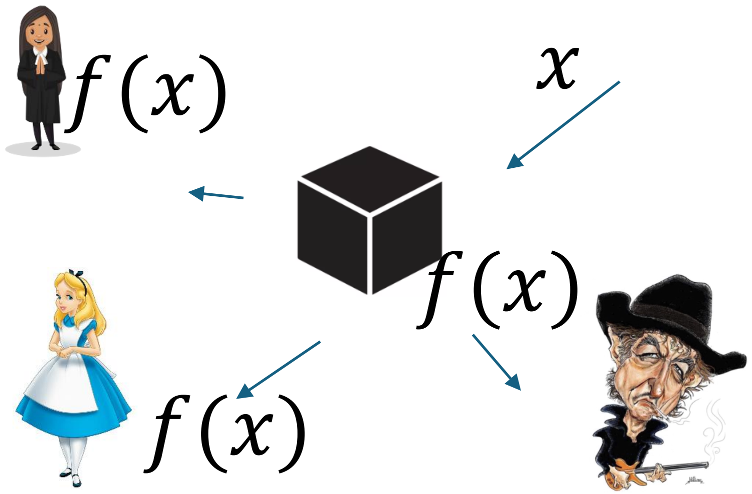
Check if the output is valid

Is f a 3-colourability for G?

Falsifiable assumption if C is efficient

Idealized Models

Assuming the existence of ideal functionalities through oracles



Replace the oracle with a real object
Hope the object behave as the ideal one

Only heuristic security

Cryptographic Hash Functions

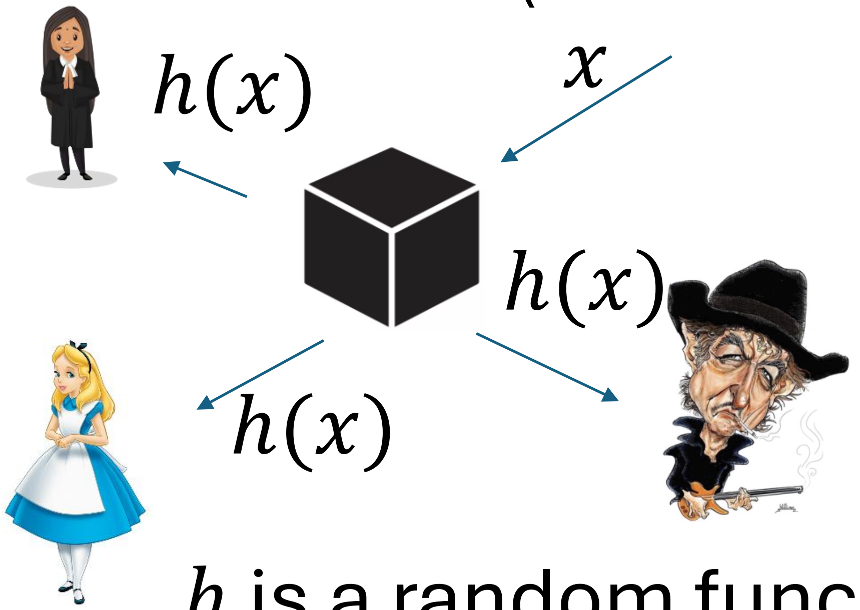
$$\mathrm{h}: D \to U$$

$x \leftarrow \$D; y = h(x)$ looks like $y \leftarrow \$U$

Real world hash

Do a bunch of shuffling
until
the output looks random
(enough)

Ideal hash (random Oracle)



h is a random function

Applications

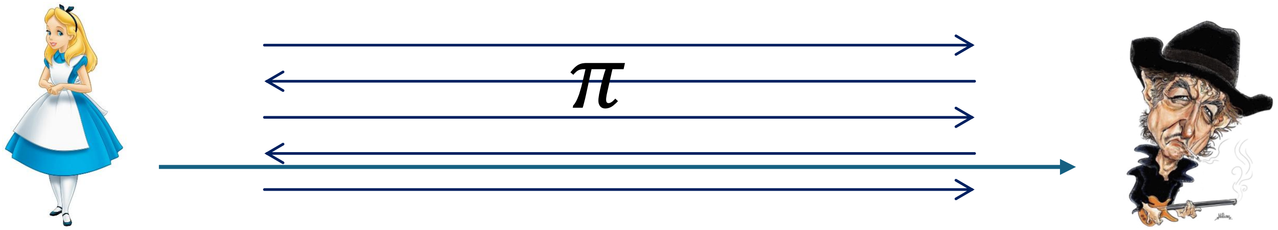
- Other cryptographic primitives
- Blockchains
- Digital currencies (Zcash)
- Electronic voting systems
- Secure and anonymous authentications
- Outsorced verifiable computation

On-line interactions and long proofs/verifications are not an option!

SNARK: Succinct Non-Interactive ARgument of Knowledge

Security Properties and Efficiency

Prover (srs, x, w) Verifier (srs, x)



- ~~Completeness: honest prover always convinces the verifier.~~
- Knowledge Soundness: if the verifier accepts, then the prover knows a valid witness.
- ~~Zero-Knowledge: the verifier learns nothing about the witness.~~

Fiat-Shamir Transform

Prover (srs, x, w)

Verifier (srs, x)

a_1

$c_1 = h(x, a_1)$

a_2 under challenge c_1

Hash h
designed such as c_1 looks random

• • •

$\pi \leftarrow (a_1, a_2, \dots)$

Check

- $\forall i \; c_i = h(x, a_1, \dots, a_{i-1})$
- Verifier $(srs, x, a_1, c_1, \dots) \rightarrow 1$

Knowledge Soundness in the Random Oracle Model

Popular Framework (Plonk, Lunar, Marlin)

- An information-theoretic proof model
  - Idealised low-degree protocols
  - Interactive Oracle Proofs
- An extractable polynomial commitment scheme
  - KZG (constant)

Compiler

Succinct interactive ZK argument

Fiat-Shamir transform

Random Oracle

SNARK

Idealized cryptographic groups (AGM, GGM)

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

Cryptographic groups

- Bracket notation for additive groups

$$\mathcal{G} = \langle g \rangle \coloneqq [1],$$
$$[x] \in \mathcal{G}: [x] = x[1] \; (= x\, g),$$

- Hardness assumptions

1. $x \leftarrow [x]$ is hard (discrete logarithm assumption)
2. $[x\, y] \leftarrow ([x], [y])$ is hard (CDH assumption)
3. $[1/\sigma] \leftarrow [\sigma]$ is hard (SDH assumption)

Generic group operations (easy)

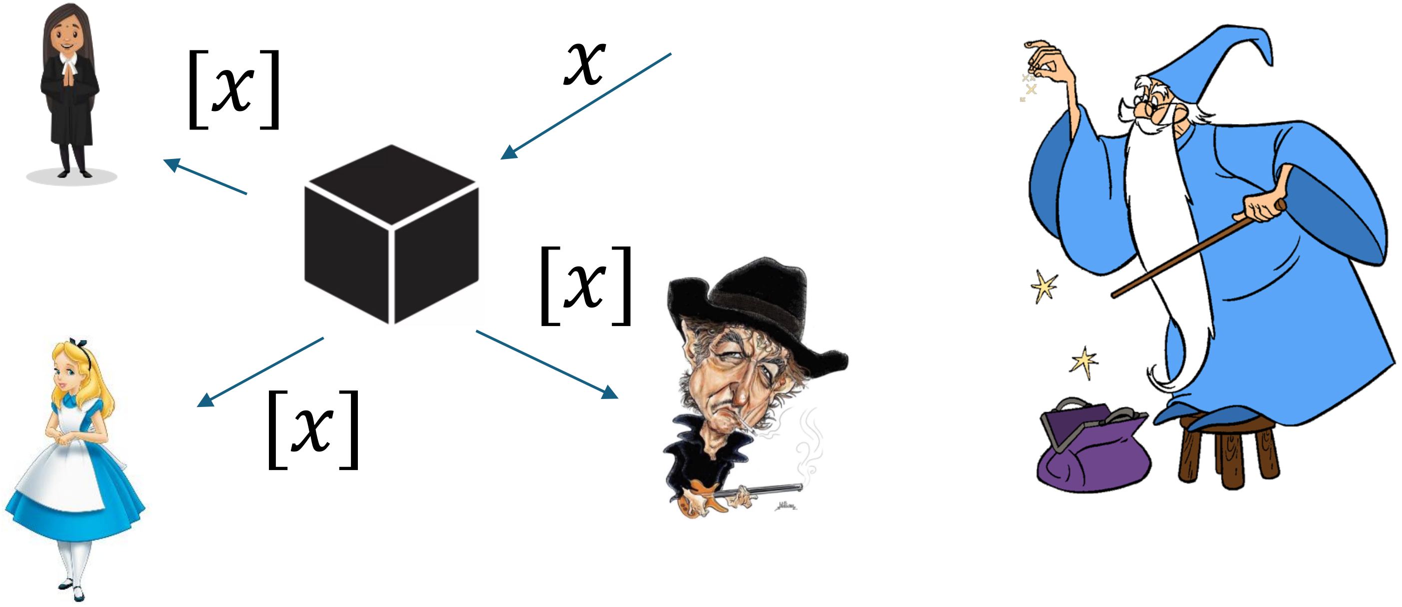
- Scalar multiplication $a[x] \to [ax]$
- Addition $[x] + [y] \to [x + y]$

Forbidden operations (hard)

- Multiplication $[x][y] \to [xy]$
- Discrete logarithm $[x] \to x$
- Inversion $[x] \to \left[\frac{1}{x}\right]$

Ideal models for Cryptographic Groups

GGM: generic group model



Perfect unstructured group.

Group elements are perfect encryptions of the exponent.

Polynomial and Rational Functions in Groups

$([1, \sigma, \ldots, \sigma^n])$

$[f(\sigma)]$

- $f(X) = \sum_{i=0}^{n} \alpha_i X^i$ poly of degree up to n
  Easy: $[f(\sigma)] = \sum_{i=0}^{n} \alpha_i [\sigma^i]$
- $f(X) = \sum_{i=0}^{m} \alpha_i X^i$ poly of degree $m > n$
  HARD: equivalent to compute $[\sigma^m]$

  Variation of CDH
- $f(X) = \frac{g(X)}{h(X)}, g, h \in Poly, h \nmid g$
  HARD: equivalent to compute $[1/\sigma]$

  Variation of SDH

Bilinear Pairing Groups

- Three additive cryptographic groups

$$(p, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, [1]_1, [1]_2, \cdot)$$

p is the order of each group

1. $[x]_1 \cdot [y]_2 = [x\, y]_T$ a trick to do one multiplication
2. $[x]_1 \leftrightarrow [x]_2$ is hard (type III pairings: no efficient isomorphism between groups)

Polynomial Commitment Scheme

- $KGen(p, n) \to ck$
- $Com(ck, f) \to C$
- $Open(ck, C, \alpha, f) \to (\eta, \pi)$
- $Verify(ck, C, \alpha, \eta, \pi) \to \{0,1\}$

Prove that $\eta = f(\alpha)$ for the committed polynomial $f(X)$ of degree $\leq n$

- Completeness:
  $Verify(ck, C, \alpha, \eta, \pi) = 1 \mid Com(ck, f) \to C \ \wedge \ Open(ck, C, \alpha, f) \to (\eta, \pi)$
- ~~Hiding:~~
  C, α, η, π does not reveal anything about f, besides that $\eta = f(\alpha)$
- Evaluation binding:
  Hard to compute two different valid openings at the same point
  $Verify(ck, C, \alpha, \eta, \pi) = Verify(ck, C, \alpha, \eta', \pi') = 1 \wedge \eta \neq \eta'$

Polynomial Commitment Scheme

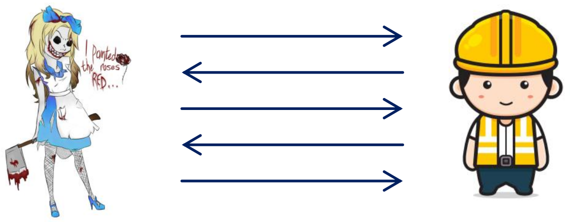
- $KGen(p, n) \rightarrow ck$
- $Com(ck, f) \rightarrow C$
- $Open(ck, C, \alpha, f) \rightarrow (\eta, \pi)$
- $Verify(ck, C, \alpha, \eta, \pi) \rightarrow \{0,1\}$

Prove the knowledge of the committed polynomial

- Black-box extraction (needed for SNARK compiler):

  Exists an extractor Ext such that for each adversary

$$A(ck) \rightarrow (C, aux), \alpha \leftarrow \$\mathbb{Z}_p$$
$$P^*(ck, C, \alpha, aux) \rightarrow (\eta, \pi) \wedge$$
$$Verify(ck, C, \alpha, \eta, \pi) = 1$$



$P^*(ck, C, \cdot, aux)$ Ext

$f(X)$
Committed polynomial

KZG Polynomial Commitment Scheme

- $KGen(p, n)$:
  $\sigma \leftarrow \$\mathbb{Z}_p$, $ck = ([\,1, \sigma, \sigma^2, \dots, \sigma^n]_1, [1, \sigma]_2)$
- $Com(ck, f)$:
  $C = [f(\sigma)]_1$
- $Open(ck, C, \alpha, f)$
  $\eta = f(\alpha), h(X) = \frac{f(X)-\eta}{X-\alpha}, \pi = [h(\sigma)]_1$
- $Verify(ck, C, \alpha, \eta, \pi) \rightarrow \{0,1\}$
  $$([f(\sigma)]_1 - \eta[1]_1) \cdot [1]_2 = [h(\sigma)]_1 \cdot ([\sigma]_2 - \alpha[1]_2)$$

Why it is secure?

$h(X) \in Poly \Leftrightarrow \eta = f(\alpha)$

KZG is black-box extractable
Assuming ideal cryptographic groups

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

GGM Criticisms

- Un-instantiability results
- Does not capture group-specific algorithms
- Reductions can always program group elements, with random known exponents

Algebraic Group Model

$[g, h, k]$

AGM: algebraic group model

$[t]$

$\alpha, \gamma, \beta : [t] = \alpha[g] + \beta[h] + \gamma[k]$

Adversaries provide a linear representation of their outputs, with respect to the group element they received on input

AGM Advantages …

- Capture some known group-specific algorithms
- Proofs by reductions

but still Criticisms

- Un-instantiability result.
- Knowledge assumptions secure in GGM/AGM but not in the standard model.

  (A given computation must pass through a specific intermediate value)

Oblivious Sampling

- Sample group elements without knowing their DL.

$$s \leftarrow \$\mathbb{Z}_p$$
$$Enc(s) = [x]$$

- DL on $Enc(D)$ is as hard as DL.

$$\Pr[Enc(s) = [x] \mid s \leftarrow \mathbb{Z}_p, x \leftarrow A([1], s)] \approx 0$$

Example: encodings on elliptic curves

Spurious Knowledge Assumptions: example

Extractor

$[1]$

$[x] \leftarrow A([1])$

x

Used by reductions

- Hold in AGM (and GGM)
- Not hold in the standard model:
    1. $s \leftarrow \$ \, \mathbb{Z}_p$
    2. $[x] = Enc(s)$

If DL holds, no extractor can compute x

Just a theoretical concern?

Interactive Plonk

Ideal Plonk

Indexer $I \rightarrow \{i_k(X)\}$

$P(I, x, w)$

$a_1(X), a_2(X)$ →

← $chall_1$

• • •

$a_{n-1}(X), a_n(X)$ →

$V(I, x)$

$$\sum_j s_j(\boldsymbol{a}(X), \boldsymbol{i}(X)) =^? 0$$

- Completeness: honest prover always convinces the verifier.
- Knowledge Soundness: if the verifier accepts, then the prover knows w.
- Zero-Knowledge: the verifier learns nothing about w.
- Succinctness: constant communication and verification complexity.

Ideal Plonk with dumb verifier

Indexer $I \to \{i_k(X)\}$

$P(I, x, w)$

$V(I, x)$

$a_1(X), a_2(X)$ →

$chall_1$ ←

• • •

$a_{m-1}(X), a_m(X)$ →

$$\xi \leftarrow \$\mathbb{Z}_p$$

$$\sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) =^? 0$$

Interactive non-optimized Plonk

SRS : $(I \rightarrow \{i_k(X)\},\ [1, \sigma, \ldots, \sigma^n]_1, [1, \sigma]_2)$

$P(SRS, x, w)$ $V(SRS, x)$

$[a_i]_1 = Com\ (a_i(X))$

$[a_1, a_2]_1$ →

$chall_1$ ←

• • •

$[a_{m-1}, a_m]_1$ →

ξ ←

$\eta_i = a_i(\xi)$

$[op_i]_1 = Open\ (a_i(X), \xi)$

$[op_1, \ldots, op_m]_1, \eta_1, \ldots, \eta_m$ →

$\forall i. Verify\ correctness\ of\ \eta_i = a_i(\xi)$

$$\sum_j s_j(\boldsymbol{\eta}, \boldsymbol{i}(\xi)) \overset{?}{=} 0$$

Linearization trick

$$\sum_j s_j\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big) d_j(X) \stackrel{?}{=} 0$$

$P(SRS, x, w)$

$V(SRS, x)$

$[a_i]_1 = Com\left(a_i(X)\right)$

$[d_i]_1 = Com\left(d_i(X)\right)$

$[\boldsymbol{a}, \boldsymbol{d}]_1$

ξ

$\eta_i = a_i(\xi)$

$[op_i]_1 = Open\left(a_i(X), \xi\right)$

$$h(X) = \sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) d_j(X)$$

$[op_h]_1 = Open\left(h(X), \xi\right)$

$[op_1, \ldots, op_m, op_h]_1$

$\eta_1, \ldots, \eta_m$

$\forall i.$ *Verify correctness of* $\eta_i = a_i(\xi)$

$$[h]_1 = \sum_j s_j\big(\boldsymbol{\eta}, \boldsymbol{i}(\xi)\big)[d_i]_1$$

Verify correctness of $0 = h(\xi)$

Batch openings (simplified description)

$P(SRS, x, w)$

$V(SRS, x)$

$[a_1]_1 = Com\left(a_1(X)\right)$

$[a_2]_1 = Com\left(a_2(X)\right)$

$[a_1, a_2]_1$

ξ

$\eta_i = a_i(\xi)$

$[op_i]_1 = Open\left(a_i(X), \xi\right)$

η_1, η_2

β

$[op]_1 = [op_1]_1 + \beta[op_2]_1$

$[op]_1$

$$[a_i - \eta_i + \beta(a_2 - \eta_2)]_1 \cdot [1]_2 \stackrel{?}{=} [op]_1 \cdot [\xi - x]_2$$

Interactive optimized Plonk

$$\sum_j s_j(\boldsymbol{a}(X), \boldsymbol{i}(X))d_j(X) \stackrel{?}{=} 0$$

$P(SRS, x, w)$ | $V(SRS, x)$

$[a_i]_1 = Com\left(a_i(X)\right)$

$[d_i]_1 = Com\left(d_i(X)\right)$

$[\boldsymbol{a}, \boldsymbol{d}]_1$

ξ

$\eta_i = a_i(\xi)$

$$h(X) = \sum_j s_j(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi))d_j(X)$$

$\eta_1, \dots, \eta_m$

β

$[op]_1$ *batch opening of* $a_i(X)$ *and* $h(X)$

$[op]_1$

- Compute commitment to $h(X)$
- Verify the correctness of all the openings with a single check

The bug: KZG Extractability

[Lipmaa,Parisella,Siim 2023]

$$KGen(p,n) \rightarrow [1,\sigma,\sigma^2,\dots,\sigma^n]_1,[1,\sigma]_2$$

$$[f(\sigma)]_1 \leftarrow A(ck,aux)$$

AGM extractor

$$\alpha_0,\alpha_1,\dots,\alpha_n : f(X) = \sum \alpha_i X^i$$

- Extraction only from commitment, without an opening
- Plonk, Lunar: SNARKs with security proof based on this assumption

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

Special Soundness for commitment schemes

$KGen(p,n) \to [1,\sigma,\sigma^2,\ldots,\sigma^n]_1,[1,\sigma]_2$

$A(ck,aux) \to (C,\{\xi_i,\eta_i,\pi_i\})$

$(C,\{\xi_i,\eta_i,\pi_i\}) \to f(X)$

If $\forall.i\ V(C,\xi_i,\eta_i,\pi_i)=1$ then $C \leftarrow Com\big(ck,f(X)\big)$ and $\forall.i\ f(\xi_i)=\eta_i$

Special soundness implies black-box extractability
but only if the commitment is opened

New security proof

[Lipmaa,Parisella,Siim 2024]

- KZG is special sound under the ARSDH assumption
- KZG is black-box extractable (but after the opening)
- Plonk (without optimizations) is knowledge-sound in ROM

No batching
No linearization trick

SNARK	Prover complexity	Verifier complexity	Proof size
Unoptimized Plonk [LPS24]	$30n\ exp$	$46\ pairings$	$23\lvert\mathbb{F}\rvert + 30\lvert\mathbb{G}_1\rvert$
Plonk	$9n\ exp$	$2\ pairings$	$6\lvert\mathbb{F}\rvert + 9\lvert\mathbb{G}_1\rvert$

Fiat-Shamir from knowledge-sound arguments

Succinct interactive ZK argument → (Fiat-Shamir transform, Random Oracle) → SNARK

Knowledge-soundness
[Gabizon,Williamson,Ciobotaru 2019]
[Lipmaa,Parisella,Siim 2024]

Loss Q^{μ}
Ignored in implementation

Special-soundness

Loss Q
Assumed in implementation

Is Plonk tightly sound in the real world?

Talk Outlines

1. Zero-knowledge and modern SNARKs.
2. Cryptographic groups.
3. The discovery of the bug.
4. A new hope.
5. Knowledge-soundness of Plonk.

Linearization trick security

$$\sum_j s_j\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big) d_j(X) =^? 0$$

$$h(X) = \sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) d_j(X)$$

$$[op_h]_1 = Open\ (h(X), \xi)$$

- Secure in AGM
- Insecure in the plain model
  [Fiore,Faonio,Russo 2024; Lipmaa,Parisella,Siim 2023]
- Knowledge-sound in AGMOS under some conditions on $d_j(X)$-s
  [Fiore,Faonio,Russo 2024]

Special-soundness of Lin-trick

The linearization trick cannot be special-sound (or knowledge-sound)
Even when knowledge-soundness holds in AGMOS

DL-assumption ⇒ Special-soundness and knowledge soundness are impossible

Important: knowledge-soundness in AGMOS is non-black-box
(adversary's random coins are given to the extractor)

Plonk use linearization trick ...

Or does it?

Linearization trick

$$\sum_j s_j\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big) d_j(X) =^? 0$$

$$h(X) = \sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) d_j(X)$$

$$[op_h]_1 = Open\ (h(X), \xi)$$

Plonk

$$\sum_j s_j\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big) d_j(X) + s\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)\ \tilde{\imath}(X) =^? 0$$

$\tilde{\imath}(X)$ public indexed polynomial

$$h(X) = \sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) d_j(X) + s\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big)\ \tilde{\imath}(X)$$

$$[op_h]_1 = Open\ (h(X), \xi)$$

RHINO

Reduction to a **h**ard assumption **i**f **no**t polynomial

$$s_1\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big) d(X) + s_2\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)\, \tilde{\imath}(X) \stackrel{?}{=} 0$$

$\tilde{\imath}(X)$ public indexed polynomial

$$d(X) = \frac{s_2\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)\, \tilde{\imath}(X)}{s_1\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)}$$

$$[1, \sigma, \sigma^2, \dots, \sigma^n]_1, [1, \sigma]_2$$

$\rightarrow$ $\boldsymbol{a}(X), [\tilde{d}]_1$

$$d(\sigma) = \tilde{d}$$

$$s_1\big(\boldsymbol{a}(\sigma), \boldsymbol{i}(\sigma)\big)[\tilde{d}] + s_2\big(\boldsymbol{a}(\sigma), \boldsymbol{i}(\sigma)\big)\, \tilde{\imath}(\sigma) \stackrel{?}{=} 0$$

RHINO

$$[1, \sigma, \sigma^2, \ldots, \sigma^n]_1, [1, \sigma]_2$$

$$\boldsymbol{a}(X), [\tilde{d}]_1$$

$$d(X) = \frac{s_2(\boldsymbol{a}(X), \boldsymbol{i}(X))\, \tilde{\imath}(X)}{s_1(\boldsymbol{a}(X), \boldsymbol{i}(X))}$$

$$[d(\sigma)]_1 = [\tilde{d}]_1$$

$$s_1(\boldsymbol{a}(\sigma), \boldsymbol{i}(\sigma))[\tilde{d}] + s_2(\boldsymbol{a}(\sigma), \boldsymbol{i}(\sigma))\, \tilde{\imath}(\sigma) \stackrel{?}{=} 0$$

- $d(X)$ is a polynomial: successfully extract the correct polynomial committed in $[\tilde{d}]_1$
- $d(X)$ is not a polynomial: HARD

Variation of SDH

Interactive Plonk is special-sound

Proof sketch:

1. KZG special-soundness $\Longrightarrow$ Extract all the polynomials $a(X)$
   - Under ARSDH KZG is special-sound [Lipmaa,Parisella,Siim 2024]
   - Batching preserves special-soundness
2. RHINO $\Longrightarrow$ Extract unopened polynomials $d(X)$
   - Under splitRSDH (variation of ARSDH, falsifiable assumption)
3. Plonk idealized protocol is special sound $\Longrightarrow$ Extract a witness
   - First time an idealized proof model is proven special-sound

Plonk is tightly knowledge-sound in the ROM

Thanks for your attention
Questions?

- **[Gabizon,Williamson,Ciobataru 2019]**
  PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive Arguments of Knowledge
- **[Lipmaa,Parisella,Siim 2023]**
  Algebraic Group Model with Oblivious Sampling
- **[Lipmaa,Parisella,Siim 2024]**
  Constant-Size zk-SNARKs in ROM from Falsifiable Assumptions
- **[Lipmaa,Parisella,Siim 2025]**
  On Knowledge-Soundness of Plonk in ROM from Falsifiable Assumptions

The ARSDH Assumption

Variant of RSDH

[González, Ràfols 2019]

Adversary A

$ck = [1, \sigma, \sigma^2, \dots, \sigma^n]_1, [1, \sigma]_2$

$S, [g, \varphi]_1$

$S \subset \mathbb{Z}_p \wedge |S| = n + 1 \wedge [g]_1 \neq [0]_1$

$$Z_S(X) := \prod_{\alpha \in S} (X - \alpha)$$

$$[g]_1 \cdot [1]_2 = [\varphi]_1 \cdot [Z_S(\sigma)]_2$$

Adaptive:
A can choose S

Rational Strong DH:

$$\varphi(X) = \frac{g(X)}{Z_S(X)}$$