On Knowledge-Soundness of Plonk in ROM from Falsifiable Assumptions

Helger Lipmaa, University of Tartu

Roberto Parisella, Simula UiB

Janno Siim, University of Tartu

Simula UiB

TARTU ÜLIKOOL · UNIVERSITY OF TARTU · 1632

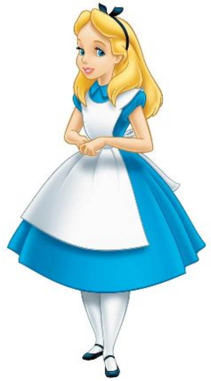
Talk Outlines

1. Plonk preliminaries and limitations in previous security proofs.
2. On the knowledge soundness of the linearization trick.
3. On the knowledge soundness of Plonk.

Ideal Plonk

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

$P(I, x, w)$



$a_1(X), a_2(X)$

$\xrightarrow{\hspace{3em}}$

$chall_1$

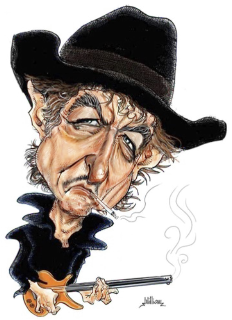
$\xleftarrow{\hspace{3em}}$

$\bullet \quad \bullet \quad \bullet$

$a_{n-1}(X), a_n(X)$

$\xrightarrow{\hspace{3em}}$

$V(I, x)$



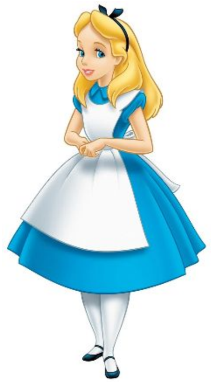
$$\sum_j s_j(\boldsymbol{a}(X), \boldsymbol{i}(X)) \stackrel{?}{=} 0$$

- 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.

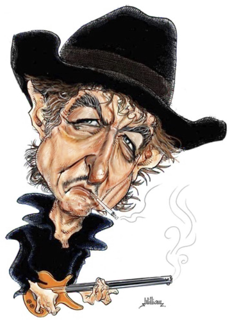
Ideal Plonk with dumb verifier

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

$P(I, x, w)$



$V(I, x)$



$a_1(X), a_2(X)$

$\longrightarrow$

$chall_1$

$\longleftarrow$

• • •

$a_{m-1}(X), a_m(X)$

$\longrightarrow$

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

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

Cryptographic groups

- Bracket notation for additive groups

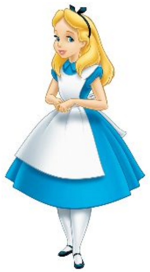
$$\mathcal{G} = \langle g \rangle := [1],$$
$$[x] \in \mathcal{G}: [x] = x[1]\ (= x\ g),$$

- Hardness assumptions

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

Polynomial and Rational Functions in Groups

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



$[f(\sigma)]$

- $f(X) = \sum_{i=0}^{n} \alpha_i X^i$ 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 $[\sigma^m]$
- $f(X) = \frac{g(X)}{h(X)}, g, h \in Poly, h \nmid g$
  HARD: equivalent to compute $[1/\sigma]$

Variation of CDH

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

KZG Polynomial Commitment Scheme

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

Why it is secure?

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

Remember the SDH assumption!

KZG has Evaluation-binding

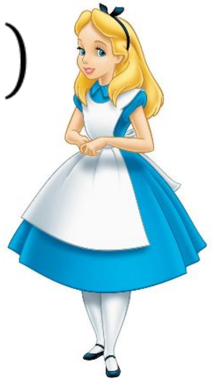
Interactive non-optimized Plonk

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

$P(SRS,x,w)$

$V(SRS,x)$

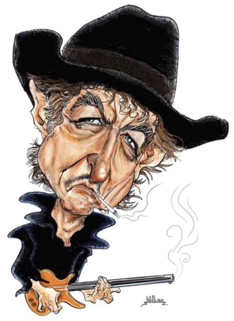
$[a_i]_1 = Com\,(a_i(X))$



$[a_1,a_2]_1$ →

$chall_1$ ←

$\bullet\ \bullet\ \bullet$



$[a_{m-1},a_m]_1$ →

ξ ←

$\eta_i = a_i(\xi)$

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

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

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

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

Linearization trick

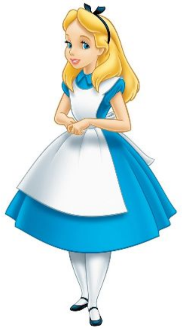
$$\sum_j s_j(\boldsymbol{a}(X), \boldsymbol{i}(X))d_j(X) \stackrel{?}{=} 0$$

$P(SRS, x, w)$

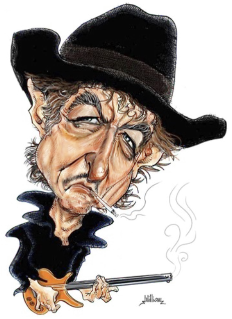
$V(SRS, x)$

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

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



ξ



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

$\eta_i = a_i(\xi)$

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

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

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

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

$\eta_1, \ldots, \eta_m$

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

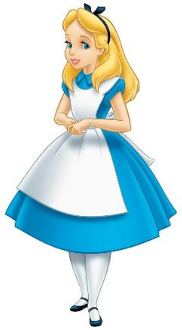
$$[h]_1 = \sum_j s_j(\boldsymbol{\eta}, \boldsymbol{i}(\xi))[d_i]_1$$

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

Batch openings (simplified description)

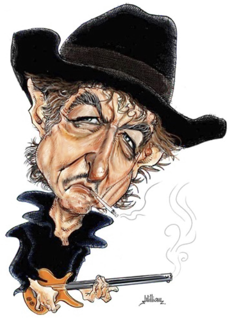
$P(SRS, x, w)$

$V(SRS, x)$



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

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



$[a_1, a_2]_1$

ξ

$\eta_i = a_i(\xi)$

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

η_1, η_2

β

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

$[op]_1$

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

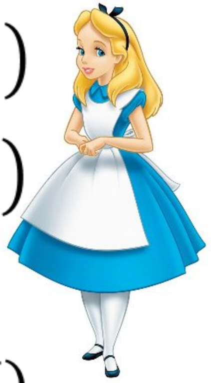
Interactive optimized Plonk $\sum_j s_j(\boldsymbol{a}(X), \boldsymbol{i}(X))d_j(X) \stackrel{?}{=} 0$

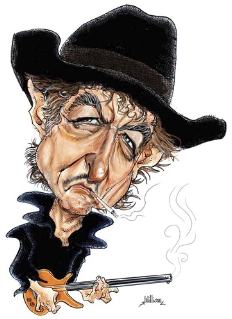
$P(SRS, x, w)$

$V(SRS, x)$

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

$[d_i]_1 = Com\,(d_i(X))$





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

$\eta_i = a_i(\xi)$

ξ

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

$\eta_1, \dots, \eta_m$

β

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

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

$[op]_1$

Popular Framework (Plonk, Lunar, Marlin)

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

Idealized cryptographic groups (AGM, GGM)

Compiler

Succinct interactive ZK argument

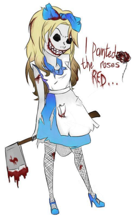
Fiat-Shamir transform

Random Oracle

SNARK

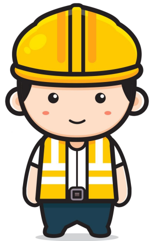
KZG Extractability

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



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

AGM
extractor



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

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

Oblivious Sampling

- Sample group elements without knowing their DL.

$$s \leftarrow D,\ \ D \text{ superpolynomial min-entropy}$$
$$Enc(s) = [x]$$

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

$$\Pr[Enc(s) = [x] \mid s \leftarrow D, x \leftarrow A([1], s)] \approx 0$$

Example: encodings on elliptic curves

Extraction from the KZG commitment does not hold in the standard model!!!

[Lipmaa,Parisella,Siim 2023]

New security proof

[Lipmaa,Parisella,Siim 2024]

- KZG is extractable under a falsifiable assumption ARSDH assumption
- New succinct knowledge-sound interactive argument Plonk PIOP (no optimization) in ROM

SNARK	Prover complexity	Verifier complexity	Proof size
Unoptimized Plonk [LPS24]			
Plonk			

No batching
No linearization trick

Fiat-Shamir from knowledge-sound arguments

Fiat-Shamir transform

Succinct interactive ZK argument

SNARK

Random Oracle

Knowledge-soundness [Gabizon,Williamson,Ciobotaru 2019]

[Lipmaa,Parisella,Siim 2024] Special-soundness

Loss Q^{μ}

Ignored in implementation

Loss Q

Assumed in implementation

[Attema,Cramer,Kohl,2021]

Is Plonk special-sound?

Talk Outlines

2. On the knowledge soundness of the linearization trick.

Linearization trick security

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

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

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

- 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
Even when knowledge-soundness holds in AGMOS

DL-assumption ⇒ Special-soundness is impossible ⇒ Knowledge-soundness is 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

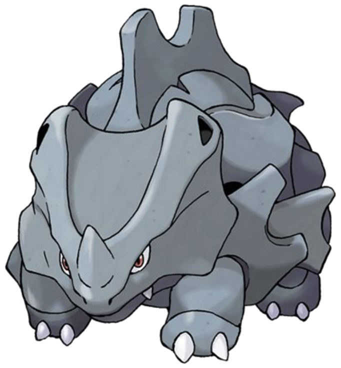
$$h(X) = \sum_j s_j\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big) d_j(X) + s\big(\boldsymbol{a}(\xi), \boldsymbol{i}(\xi)\big)\, \tilde{\imath}(X)$$

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

Talk Outlines

3. On the knowledge soundness of Plonk.

RHINO

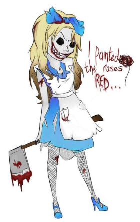


Reduction to a **h**ard assumption **i**f **no**t polynomial

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

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

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



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

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

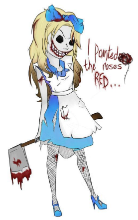
$$d(X) = \frac{s_2\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)\, \tilde{\imath}(X)}{s_1\big(\boldsymbol{a}(X), \boldsymbol{i}(X)\big)}$$

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

RHINO

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

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



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

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

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

- $d(X)$ 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
- **[Attema,Cramer,Kohl,2021]**
  A compressed Σ-protocol theory for lattices
- **[Lipmaa,Parisella,Siim 2023]**
  Algebraic Group Model with Oblivious Sampling
- **[Lipmaa,Parisella,Siim 2024]**
  Constant-Size zk-SNARKs in ROM from Falsifiable Assumptions
- **[Fiore,Faonio,Russo 2024]**
  Real-world Universal zkSNARKs are non-malleable

The splitRSDH Assumption

Variant of ARSDH

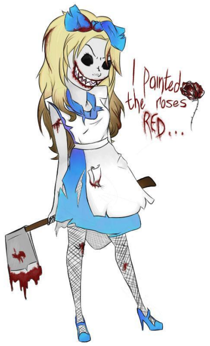
[Lipmaa,Parisella,Siim 2024]

Adversary A

Public polynomials $\psi_i(X)$

$\max \deg \psi_i(X) = n_\psi$

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



$\boldsymbol{S}, L(X), [\tilde{\boldsymbol{d}}, \psi]_1$

$$S \subset \mathbb{Z}_p \wedge |S| = n_S > n + n_\psi + 1 \wedge Z_S(X) := \prod_{\alpha \in S}(X - \alpha)$$

$$n + n_\psi < \deg L(X) < n_S$$

$$\sum \left[\tilde{d}_i \psi_i(\sigma)\right]_1 = [\psi Z_S(\sigma)]_1 + [L(\sigma)]_1$$

$\leq n_\psi + n$

$\geq n_S$

Variation of SDH