How (Not) to Simulate PLONK

https://ia.cr/2024/848

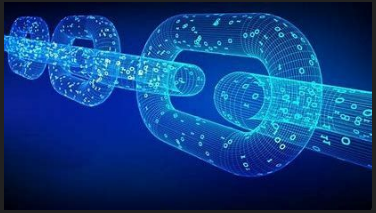
Marek Sefranek
TU Wien

Motivation

- Zero-knowledge: prove something is true without revealing why
  - For example: prove age over certain limit (“digital ID”) without revealing it
  - Comply with rules with minimal disclosure of information (GDPR)

Motivation

- Zero-knowledge: prove something is true without revealing why
  - For example: prove age over certain limit (“digital ID”) without revealing it
  - Comply with rules with minimal disclosure of information (GDPR)
- Applications:
  - Enforce parties follow a protocol (MPC)
  - Verifiable computation, anonymous credentials
  - Enable trust in decentralized systems such as blockchains
  - Fully anonymous cryptocurrencies, e.g. Zcash
  - …



TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Zero-Knowledge Proof

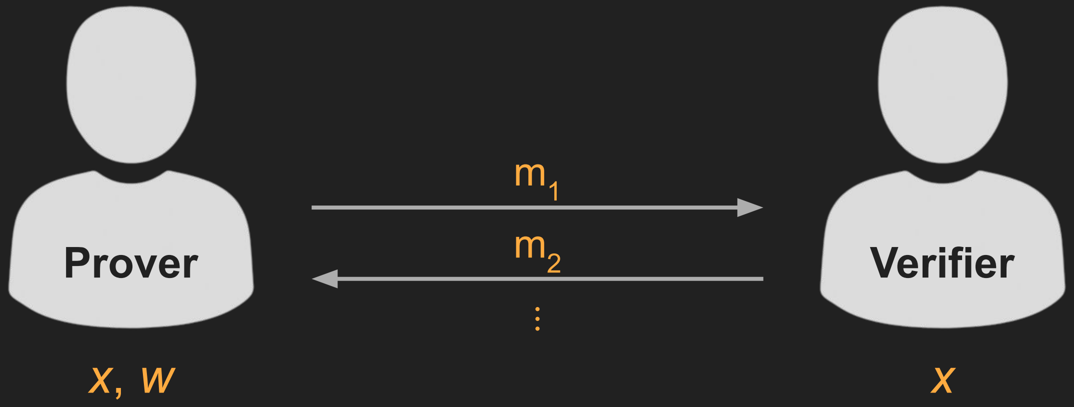
- Let *R* be an NP relation and *L* the corresponding language

Zero-Knowledge Proof

- Let R be an NP relation and L the corresponding language
- Prove statement $x \in L$ without revealing witness w

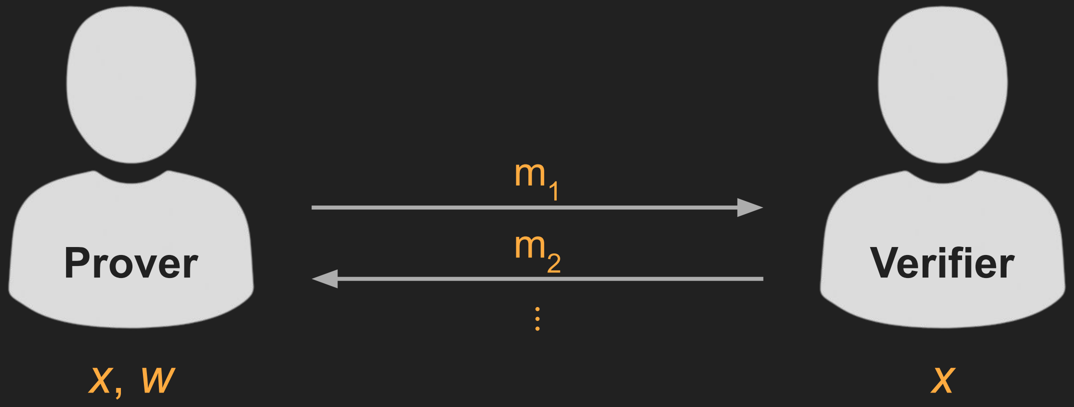
Zero-Knowledge Proof

- Let R be an NP relation and L the corresponding language
- Prove statement $x \in L$ without revealing witness w



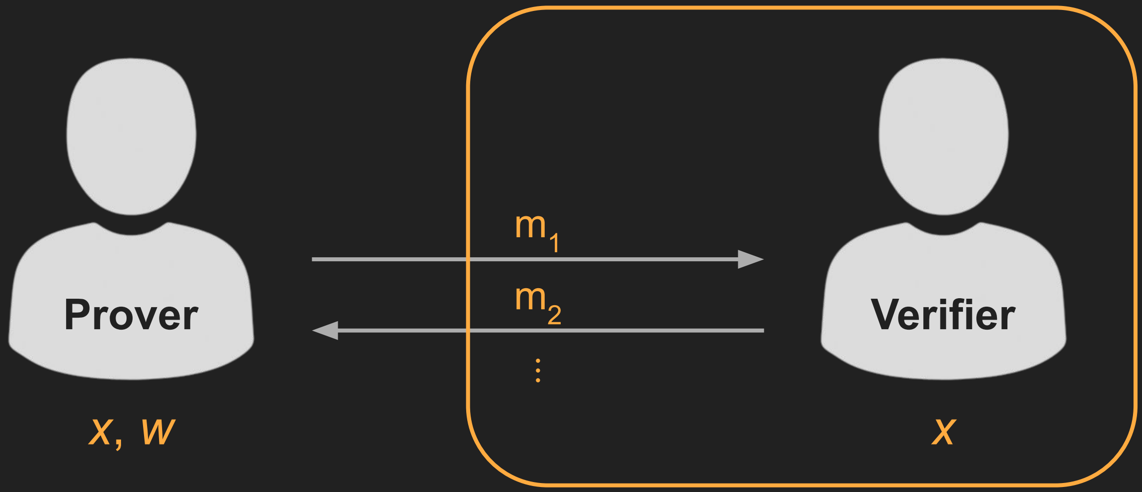
Zero-Knowledge Proof – Properties

- Completeness/Soundness: statement true ⟺ verifier accepts



Zero-Knowledge Proof – Properties

- Completeness/Soundness: statement true ⟺ verifier accepts
- Zero Knowledge: can efficiently simulate view of verifier only given x

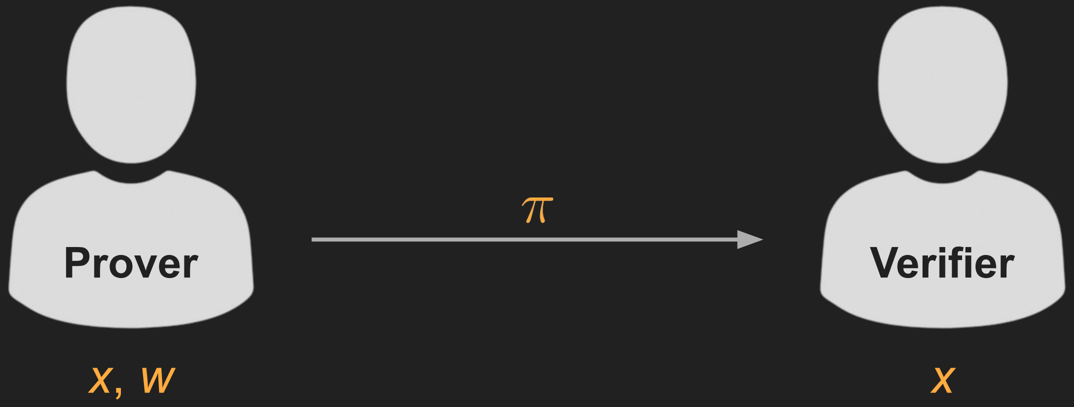


zk-SNARK

- Zero-Knowledge Succinct Non-interactive ARgument of Knowledge

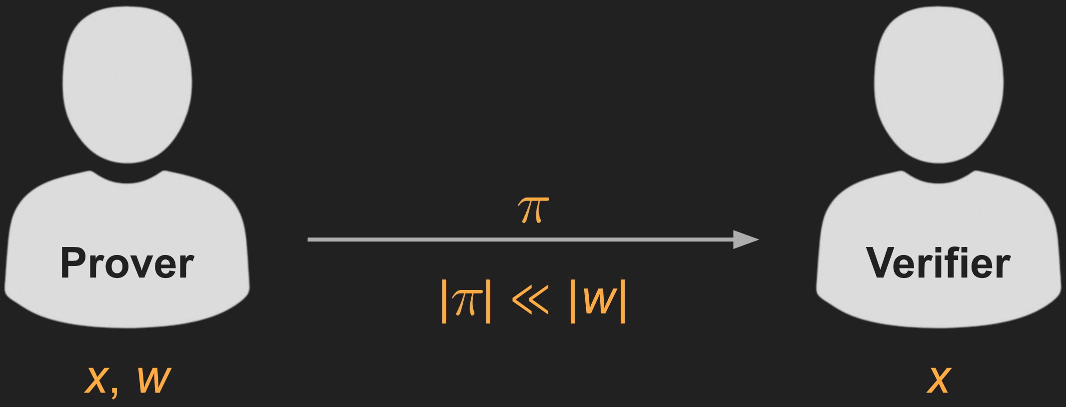
zk-SNARK

- Zero-Knowledge Succinct Non-interactive ARgument of Knowledge



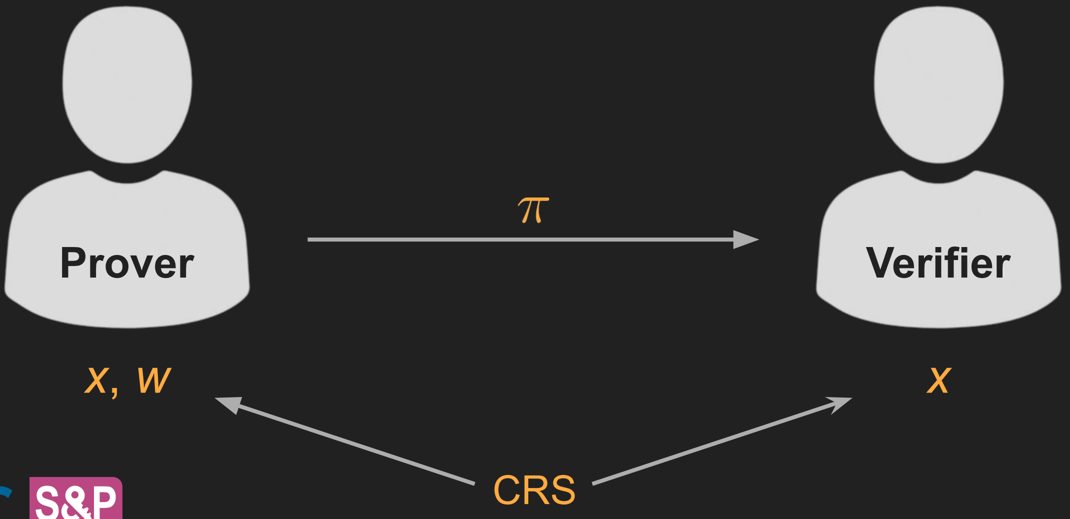
zk-SNARK

- Zero-Knowledge Succinct Non-interactive ARgument of Knowledge



zk-SNARK

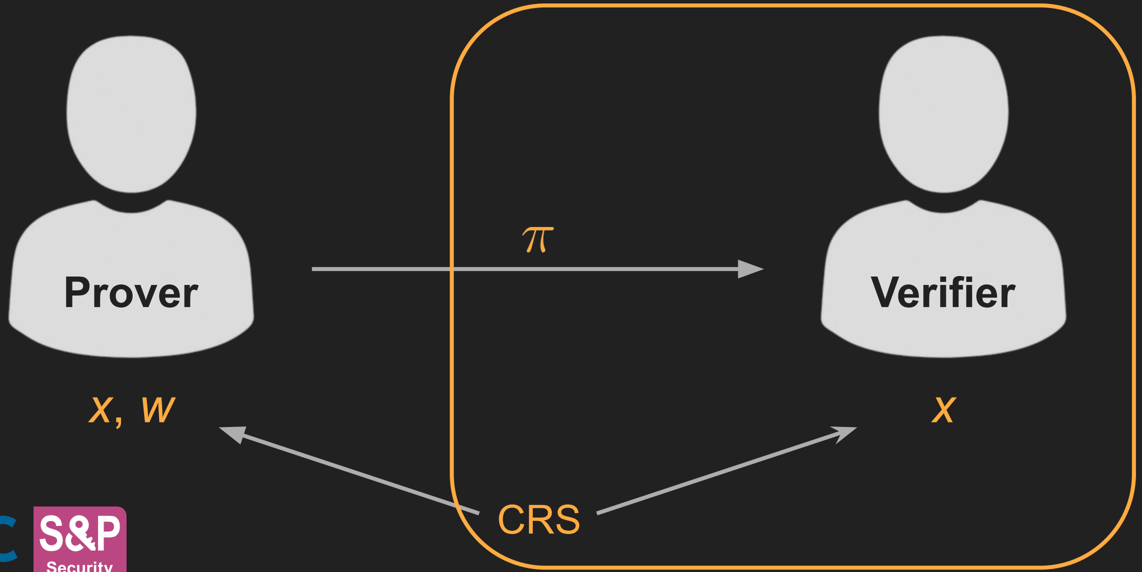
- Zero-Knowledge Succinct Non-interactive ARgument of Knowledge
- Need trusted setup: common reference string (CRS)



TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

zk-SNARK

- Zero-Knowledge Succinct Non-interactive ARgument of Knowledge
- Need trusted setup: common reference string (CRS)
- Zero knowledge: simulator can set up CRS, knowing “trapdoor”



PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]

PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]
- A proof is ≈0.5 kB and can be verified in milliseconds

PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]
- A proof is ≈0.5 kB and can be verified in milliseconds
- Universal & updatable structured reference string (SRS)

PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]
- A proof is ≈0.5 kB and can be verified in milliseconds
- Universal & updatable structured reference string (SRS)
- Knowledge sound in AGM + ROM (or just ROM [LPS24])

PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]
- A proof is ≈0.5 kB and can be verified in milliseconds
- Universal & updatable structured reference string (SRS)
- Knowledge sound in AGM + ROM (or just ROM [LPS24])
- Supports custom gates and lookup gates

PLONK

- State-of-the-art zk-SNARK by Gabizon, Williamson & Ciobotaru [GWC19]
- A proof is ≈0.5 kB and can be verified in milliseconds
- Universal & updatable structured reference string (SRS)
- Knowledge sound in AGM + ROM (or just ROM [LPS24])
- Supports custom gates and lookup gates
- Deployed in a variety of real-world projects

Aztec polygon Espresso Matter Labs DUSK

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

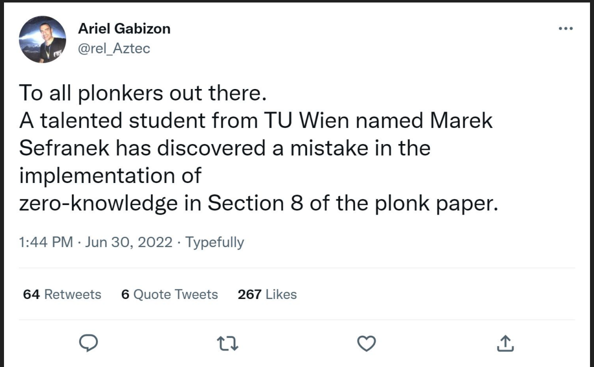
Main Contribution

- But no proof that PLONK is zero-knowledge!

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Main Contribution

- But no proof that PLONK is zero-knowledge!
- Found vulnerability in its ZK implementation & proposed fix


Ariel Gabizon
@rel_Aztec

To all plonkers out there.
A talented student from TU Wien named Marek Sefranek has discovered a mistake in the implementation of
zero-knowledge in Section 8 of the plonk paper.

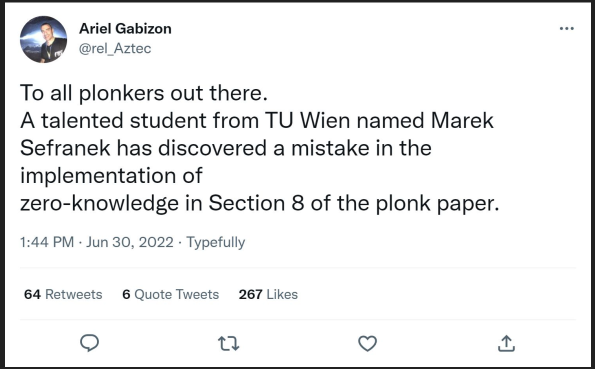
1:44 PM · Jun 30, 2022 · Typefully

64 Retweets 6 Quote Tweets 267 Likes

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Main Contribution

- But no proof that PLONK is zero-knowledge!
- Found vulnerability in its ZK implementation & proposed fix



- Formal security proof that it now achieves statistical ZK

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

KZG Polynomial Commitment [KZG10]

- Succinctly commit to a polynomial $f \in \mathbb{F}[X]$

KZG Polynomial Commitment [KZG10]

- Succinctly commit to a polynomial $f \in \mathbb{F}[X]$
- Later prove evaluations, i.e., for any point $x \in \mathbb{F}$ show that $f(x) = y$

KZG Polynomial Commitment [KZG10]

- Succinctly commit to a polynomial $f \in \mathbb{F}[X]$
- Later prove evaluations, i.e., for any point $x \in \mathbb{F}$ show that $f(x) = y$
- SRS: $\left(g_1, g_1^{\tau}, g_1^{\tau^2}, \ldots, g_1^{\tau^d}, g_2, g_2^{\tau}\right)$ for uniform “trapdoor” $\tau \in \mathbb{F}$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

KZG Polynomial Commitment [KZG10]

- Succinctly commit to a polynomial $f \in \mathbb{F}[X]$
- Later prove evaluations, i.e., for any point $x \in \mathbb{F}$ show that $f(x) = y$
- SRS: $\left(g_1, g_1^{\tau}, g_1^{\tau^2}, \ldots, g_1^{\tau^d}, g_2, g_2^{\tau}\right)$ for uniform "trapdoor" $\tau \in \mathbb{F}$
- A commitment to a polynomial $f(X) = \sum_{i=0}^{d} f_i X^i \in \mathbb{F}[X]$ is

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

KZG Polynomial Commitment [KZG10]

- Succinctly commit to a polynomial $f \in \mathbb{F}[X]$
- Later prove evaluations, i.e., for any point $x \in \mathbb{F}$ show that $f(x) = y$
- SRS: $\left(g_1, g_1^{\tau}, g_1^{\tau^2}, \ldots, g_1^{\tau^d}, g_2, g_2^{\tau}\right)$ for uniform "trapdoor" $\tau \in \mathbb{F}$
- A commitment to a polynomial $f(X) = \sum_{i=0}^{d} f_i X^i \in \mathbb{F}[X]$ is

$$c := \prod_{i=0}^{d} \left(g_1^{\tau^i}\right)^{f_i} = g_1^{\sum_{i=0}^{d} f_i \tau^i} = g_1^{f(\tau)}$$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

PLONK – Simplified Overview

- For $Z(X) := (X - \omega^1)(X - \omega^2)\cdots(X - \omega^n)$, want to show $Z(X) \mid C(X)$

PLONK – Simplified Overview

- For $Z(X) := (X - \omega^1)(X - \omega^2)\cdots(X - \omega^n)$, want to show $Z(X) \mid C(X)$
- Prover commits to $C(X)$ and quotient polynomial $T(X)$ [KZG10]

PLONK – Simplified Overview

- For $Z(X) := (X - \omega^1)(X - \omega^2)\cdots(X - \omega^n)$, want to show $Z(X) \mid C(X)$
- Prover commits to $C(X)$ and quotient polynomial $T(X)$ [KZG10]
- Its degree is $3n$, where n is the number of gates

PLONK – Simplified Overview

- For $Z(X) := (X - \omega^1)(X - \omega^2)\cdots(X - \omega^n)$, want to show $Z(X) \mid C(X)$
- Prover commits to $C(X)$ and quotient polynomial $T(X)$ [KZG10]
- Its degree is $3n$, where n is the number of gates
- Other polynomials have degree $n \Longrightarrow$ SRS has to be 3x as long

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

PLONK – Simplified Overview

- For $Z(X) := (X - \omega^1)(X - \omega^2)\cdots(X - \omega^n)$, want to show $Z(X) \mid C(X)$
- Prover commits to $C(X)$ and quotient polynomial $T(X)$ [KZG10]
- Its degree is $3n$, where n is the number of gates
- Other polynomials have degree $n \implies$ SRS has to be 3x as long
- To avoid this, PLONK splits T into 3 degree-n polynomials T_1, T_2, T_3 s.t.

$$T(X) = T_1(X) + X^n\, T_2(X) + X^{2n}\, T_3(X)$$

PLONK – Proof

$$\pi_{\text{PLONK}} := \begin{pmatrix} g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, \\ A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \end{pmatrix}$$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

PLONK – Proof

KZG commitments to witness polynomials

$$\pi_{\text{PLONK}} := \left(g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, \atop A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \right)$$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

PLONK – Proof

KZG commitments to witness polynomials

KZG commitments to split quotient polynomial

$$\pi_{\text{PLONK}} := \left(\begin{array}{c} g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, \\ A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \end{array} \right)$$

PLONK – Proof

KZG commitments to witness polynomials

KZG commitments to split quotient polynomial

Batched KZG opening proofs

$$\pi_{\text{PLONK}} := \left(\begin{array}{c} g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, \\ A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \end{array} \right)$$

PLONK – Proof

KZG commitments to witness polynomials

KZG commitments to split quotient polynomial

Batched KZG opening proofs

$$\pi_{\text{PLONK}} := \left(g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \right)$$

Witness polynomials evaluated at challenge

PLONK – Proof

KZG commitments to witness polynomials

KZG commitments to split quotient polynomial

Batched KZG opening proofs

$$\pi_{\text{PLONK}} := \Big(g_1^{A(\tau)}, g_1^{B(\tau)}, g_1^{C(\tau)}, g_1^{\Phi(\tau)}, g_1^{T_1(\tau)}, g_1^{T_2(\tau)}, g_1^{T_3(\tau)}, g_1^{Q_1(\tau)}, g_1^{Q_2(\tau)}, A(\delta), B(\delta), C(\delta), \Phi(\delta\omega), S_{\sigma,1}(\delta), S_{\sigma,2}(\delta) \Big)$$

Witness polynomials evaluated at challenge

Evaluations of public polynomials

Zero Knowledge Vulnerability

- Without splitting $T(X)$:
  - Can be simulated as $T(\tau)$ can be computed given the KZG trapdoor τ
  - Proof independent of witness

Zero Knowledge Vulnerability

- Without splitting $T(X)$:
  - Can be simulated as $T(\tau)$ can be computed given the KZG trapdoor τ
  - Proof independent of witness
- With the optimization:
  - T_1, T_2, T_3 leak too much information about $T(X)$
  - Proof no longer independent of witness!

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + X^n\ T_2(X) + X^{2n}\ T_3(X)$$

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + r_1 X^n + X^n (T_2(X) - r_1 \qquad) + X^{2n} \; T_3(X)$$

for randomly chosen $r_1 \qquad \in \mathbb{F}$

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + r_1 X^n + X^n (T_2(X) - r_1 + r_2 X^n) + X^{2n} (T_3(X) - r_2)$$

for randomly chosen $r_1, r_2 \in \mathbb{F}$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + r_1 X^n + X^n (T_2(X) - r_1 + r_2 X^n) + X^{2n} (T_3(X) - r_2)$$

for randomly chosen $r_1, r_2 \in \mathbb{F}$

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + r_1 X^n + X^n (T_2(X) - r_1 + r_2 X^n) + X^{2n} (T_3(X) - r_2)$$

  for randomly chosen $r_1, r_2 \in \mathbb{F}$
- Can now be simulated as the value $T(\tau)$ can be:
  1. Choose uniform values for $T_2(\tau)$ and $T_3(\tau)$
  2. Set $T_1(\tau) := T(\tau) - \tau^n T_2(\tau) - \tau^{2n} T_3(\tau)$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Zero Knowledge Fix

- Randomize T_1, T_2, T_3 so they are uniform conditioned on satisfying

$$T(X) = T_1(X) + r_1 X^n + X^n (T_2(X) - r_1 + r_2 X^n) + X^{2n} (T_3(X) - r_2)$$

for randomly chosen $r_1, r_2 \in \mathbb{F}$

- Can now be simulated as the value $T(\tau)$ can be:
  1. Choose uniform values for $T_2(\tau)$ and $T_3(\tau)$
  2. Set $T_1(\tau) := T(\tau) - \tau^n T_2(\tau) - \tau^{2n} T_3(\tau)$
- Preserves knowledge soundness as verifier remains the same!

Attack on Old PLONK

- “Old PLONK not stat. witness indistinguishable” $\Longrightarrow$ “not stat. ZK”

Attack on Old PLONK

- “Old PLONK not stat. witness indistinguishable” $\Longrightarrow$ “not stat. ZK”
- Idea: Solve system of linear equations to recover blinding scalars used by prover to mask witness polynomials

Attack on Old PLONK

- “Old PLONK not stat. witness indistinguishable” $\implies$ “not stat. ZK”
- Idea: Solve system of linear equations to recover blinding scalars used by prover to mask witness polynomials
- Compare against resulting values of $T_1(\tau)$, $T_2(\tau)$, $T_3(\tau)$
  1. If correct witness is used, check will always pass
  2. Otherwise, check will fail w.h.p.

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Attack on Old PLONK

- "Old PLONK not stat. witness indistinguishable" $\Longrightarrow$ "not stat. ZK"
- Idea: Solve system of linear equations to recover blinding scalars used by prover to mask witness polynomials
- Compare against resulting values of $T_1(\tau)$, $T_2(\tau)$, $T_3(\tau)$
  1. If correct witness is used, check will always pass
  2. Otherwise, check will fail w.h.p.
- For example:
  - Prover picks random $\rho_1, \rho_2 \in \mathbb{F}$ and defines $A(X) := (\rho_1 X + \rho_2)\, Z(X) + \sum_{i \in [n]} w_i\, L_i(X)$
  - Proof reveals $A(\tau)$, $A(\delta)$ $\Longrightarrow$ system of 2 linear equations in 2 unknowns

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

More in the Full Paper…

- Proof of statistical (computational) ZK in ROM (collision-resistant *H*)
- Unbounded attack on witness indistinguishability of old PLONK

https://ia.cr/2024/848

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

More in the Full Paper…

- Proof of statistical (computational) ZK in ROM (collision-resistant *H*)
- Unbounded attack on witness indistinguishability of old PLONK

https://ia.cr/2024/848

Thanks!

Questions?

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

References

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge. Cryptology ePrint Archive, Paper 2019/953, 2019. https://eprint.iacr.org/2019/953.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In Advances in Cryptology – ASIACRYPT 2010, volume 6477 of LNCS, pages 177–194. Springer, 2010. https://doi.org/10.1007/978-3-642-17373-8_11.

[LPS24] Helger Lipmaa, Roberto Parisella, and Janno Siim. On Knowledge-Soundness of Plonk in ROM from Falsifiable Assumptions. Cryptology ePrint Archive, Paper 2024/994, 2024. https://eprint.iacr.org/2024/994.