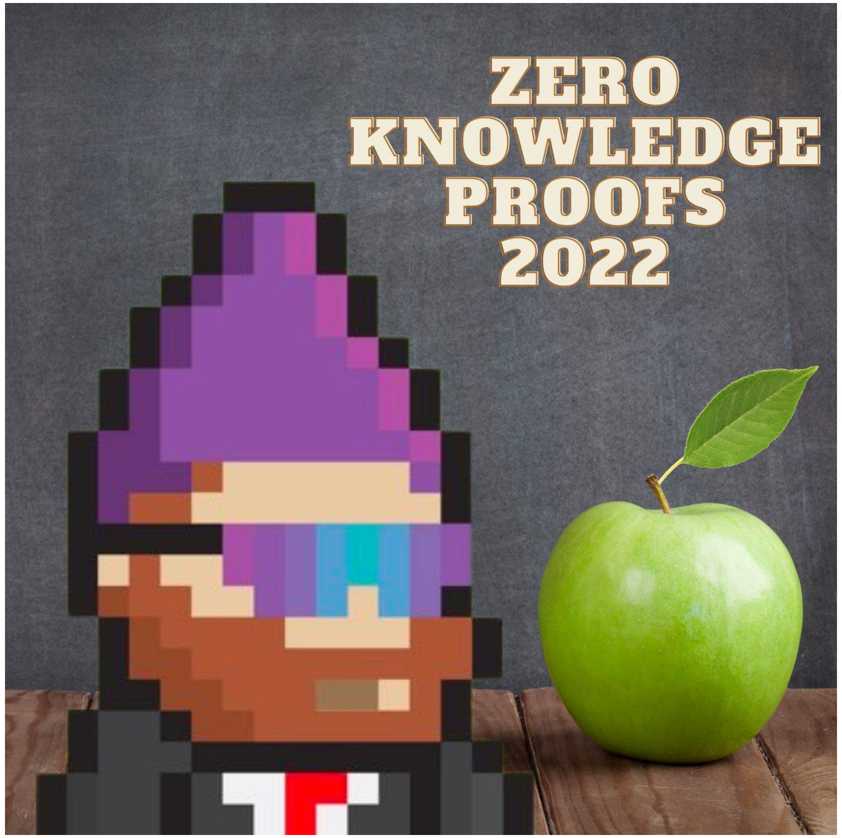
ZK Seminar for SpearbitDAO

Porter Adams (@cyber_porter)

Why am I talking to you

- Blockchain security researcher at Kudelski Security
- Started learning cryptography in 2013
- Taught a ZK class to 1600 people from twitter



KUDELSKI SECURITY

What is ZK?

- Zero Knowledge **Proofs**

If you hear “Zero Knowledge”,

Think “Verifiable Computation”

What is ZK?

- Zero Knowledge **Proofs**

If you hear “Zero Knowledge”,

Think “Verifiable Computation”

- ZK Rollups
- Tornado Cash
- Dark Forest

Buzzwords

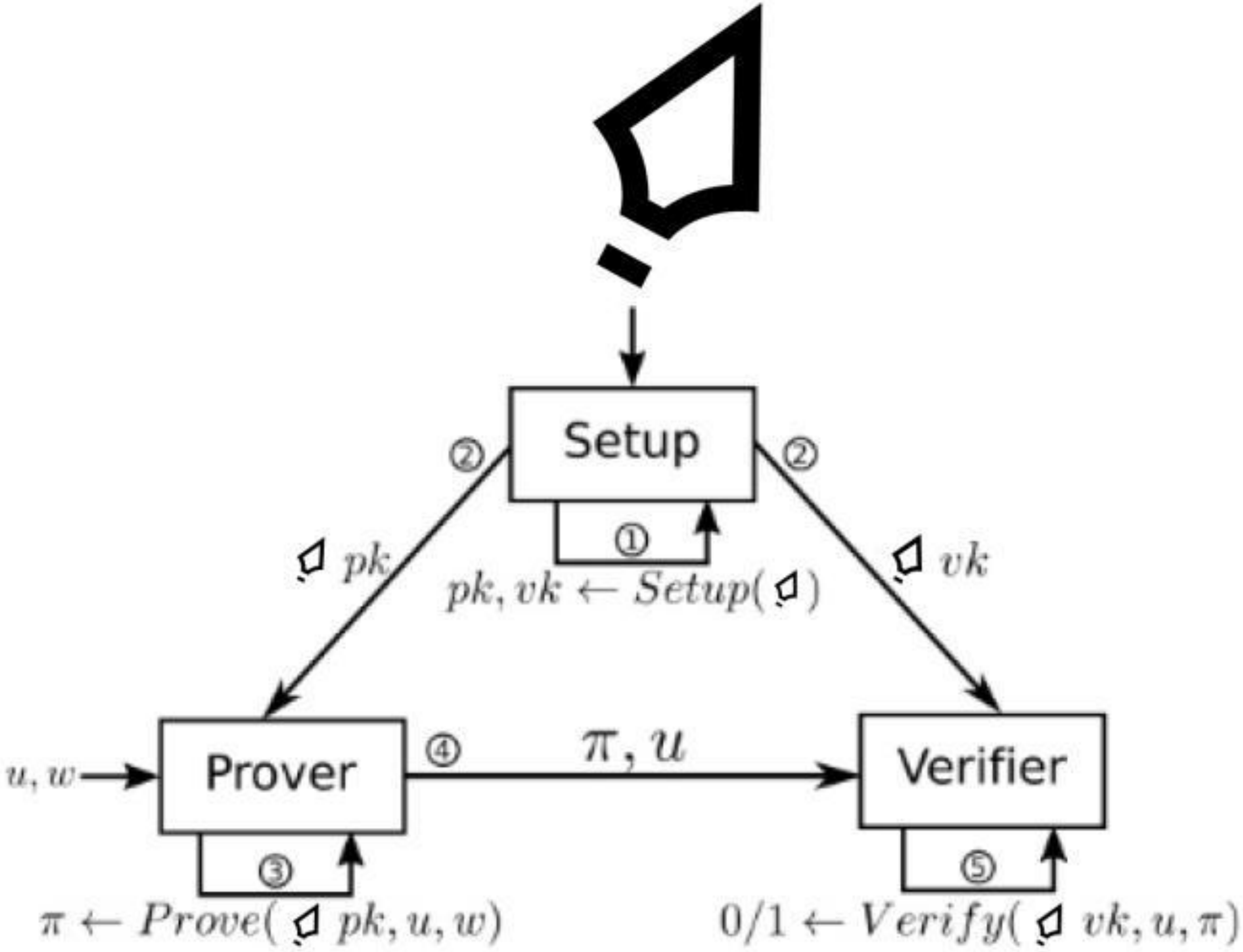
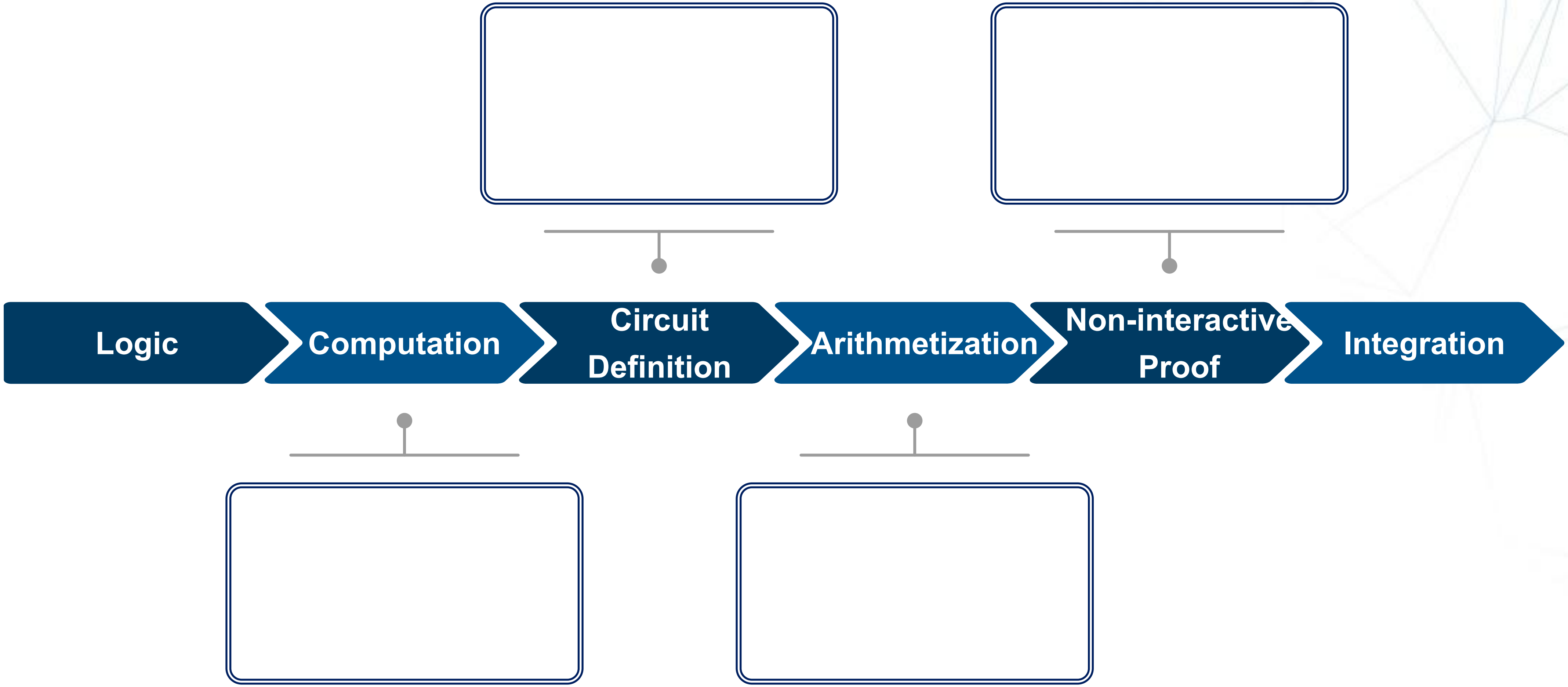


Figure 1: Spearbit Knowledge Proof System

More buzzwords



Logic

Computation

Circuit Definition

Arithmetization

Non-interactive Proof

Integration

ZK example: Scroll

- Scroll

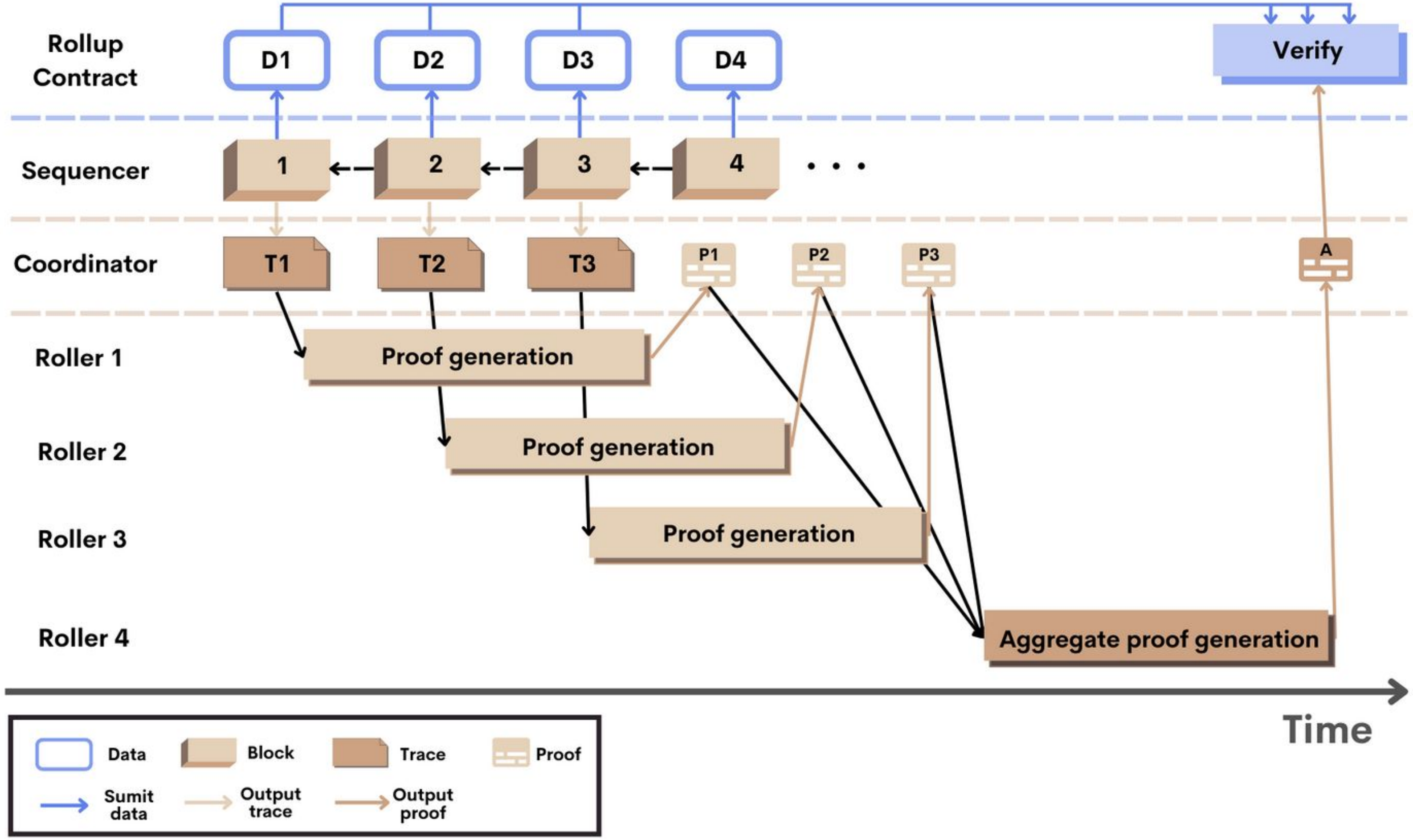
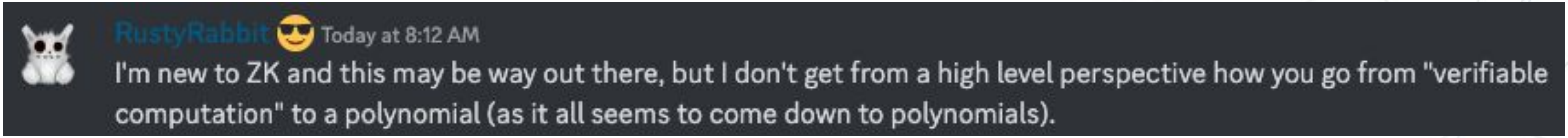


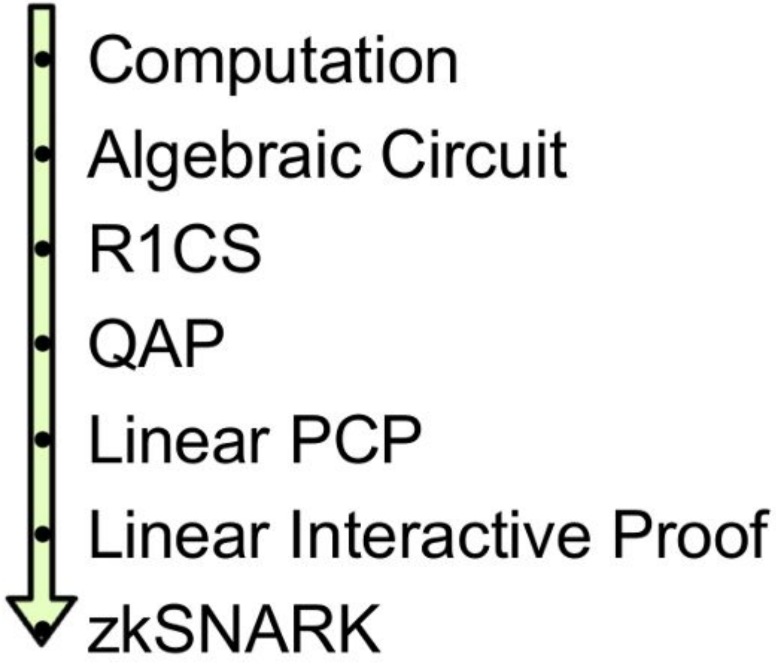
Figure 3. Scroll workflow

Q&A



RustyRabbit 😎 Today at 8:12 AM

I'm new to ZK and this may be way out there, but I don't get from a high level perspective how you go from "verifiable computation" to a polynomial (as it all seems to come down to polynomials).



Computation

Algebraic Circuit

R1CS

QAP

Linear PCP

Linear Interactive Proof

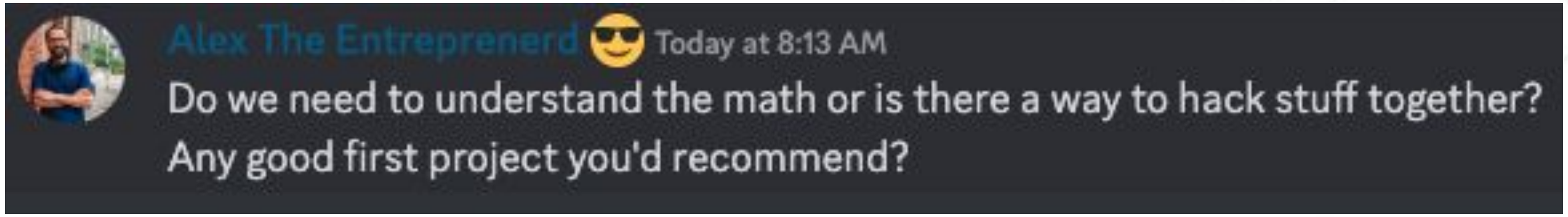
zkSNARK

Vitalik Buterin

Dec 11, 2016 · 13 min read · Listen

Quadratic Arithmetic Programs: from Zero to Hero

Q&A

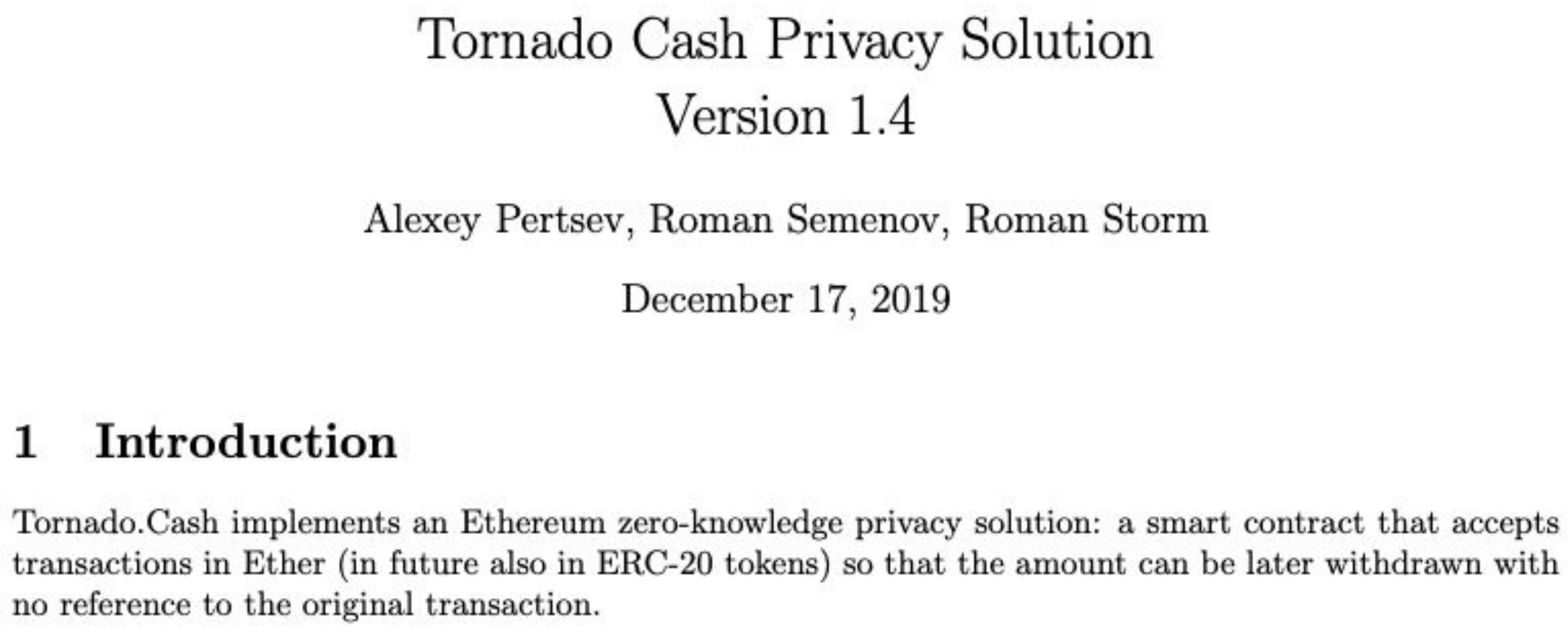


Alex The Entreprenerd 😎 Today at 8:13 AM

Do we need to understand the math or is there a way to hack stuff together? Any good first project you'd recommend?

ZK example: Tornado Cash

Test yourself: Read the Tornado Cash whitepaper (only 3 pages)



Tornado Cash Privacy Solution
Version 1.4

Alexey Pertsev, Roman Semenov, Roman Storm

December 17, 2019

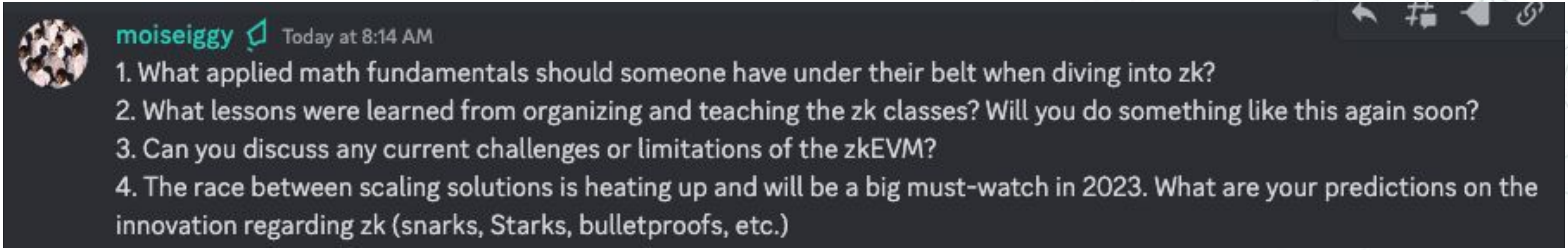
1 Introduction

Tornado.Cash implements an Ethereum zero-knowledge privacy solution: a smart contract that accepts transactions in Ether (in future also in ERC-20 tokens) so that the amount can be later withdrawn with no reference to the original transaction.

ZK Exercises

- Read Tornado Cash whitepaper
  - https://berkeley-defi.github.io/assets/material/Tornado%20Cash%20Whitepaper.pdf
- ZK Battleship
  - https://github.com/tommymsz006/zkbattleship
- Mina Protocol:
  - https://minaprotocol.com/get-started
- Aleo Leo examples
  - https://developer.aleo.org/leo/examples

Q&A



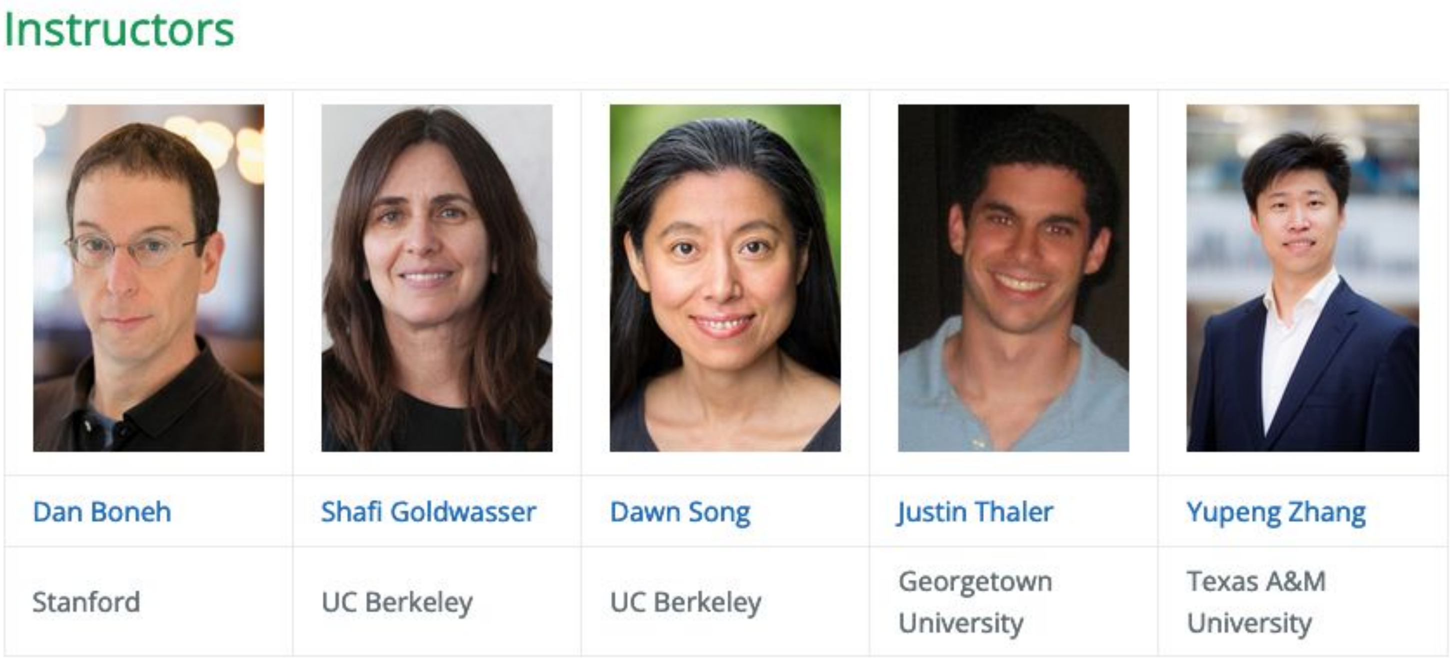
moiseiggy Today at 8:14 AM

1. What applied math fundamentals should someone have under their belt when diving into zk?
2. What lessons were learned from organizing and teaching the zk classes? Will you do something like this again soon?
3. Can you discuss any current challenges or limitations of the zkEVM?
4. The race between scaling solutions is heating up and will be a big must-watch in 2023. What are your predictions on the innovation regarding zk (snarks, Starks, bulletproofs, etc.)

KUDELSKI SECURITY

If you really want to learn ZK Math…

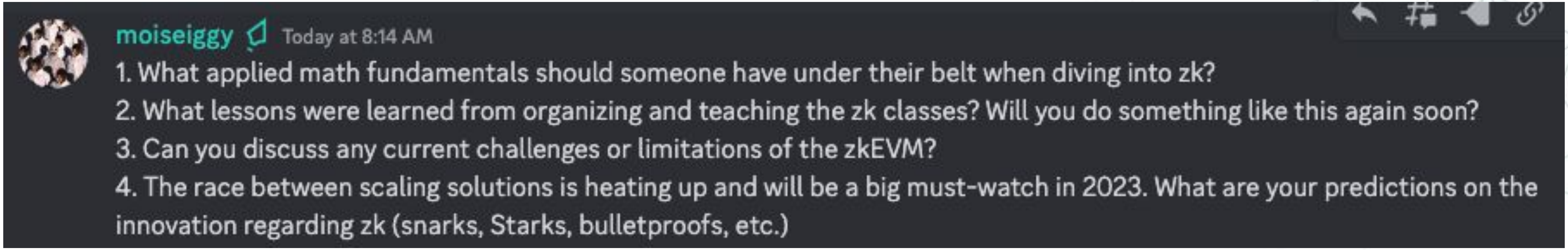
- Need one ~college class in:
  - Number Theory
  - Abstract Algebra (groups, finite fields)
- Take the ZK MOOC:
  - https://zk-learning.org/



Instructors

Dan Boneh	Shafi Goldwasser	Dawn Song	Justin Thaler	Yupeng Zhang
Stanford	UC Berkeley	UC Berkeley	Georgetown University	Texas A&M University

Q&A



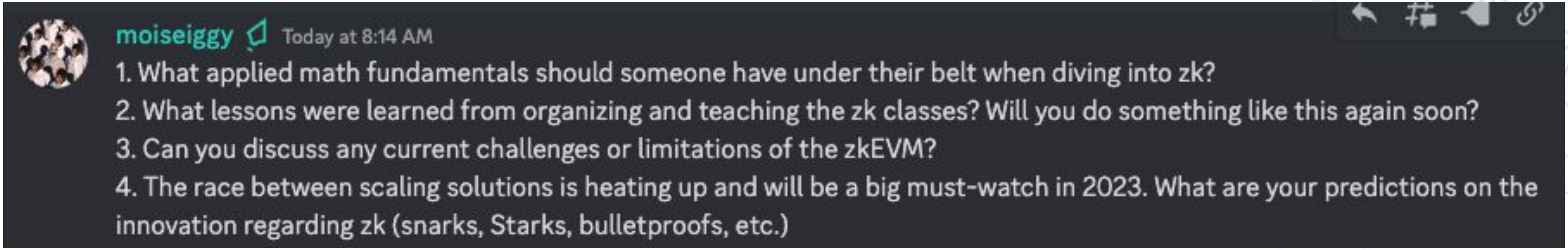
moiseiggy Today at 8:14 AM

1. What applied math fundamentals should someone have under their belt when diving into zk?
2. What lessons were learned from organizing and teaching the zk classes? Will you do something like this again soon?
3. Can you discuss any current challenges or limitations of the zkEVM?
4. The race between scaling solutions is heating up and will be a big must-watch in 2023. What are your predictions on the innovation regarding zk (snarks, Starks, bulletproofs, etc.)

My lessons from teaching zk

- Creating good zk content takes a lot of work
  - In particular, I don’t think any zk classes are quick and well-explained
  - Ideally, someone could ramp up to zk in 1-2 hours

Q&A



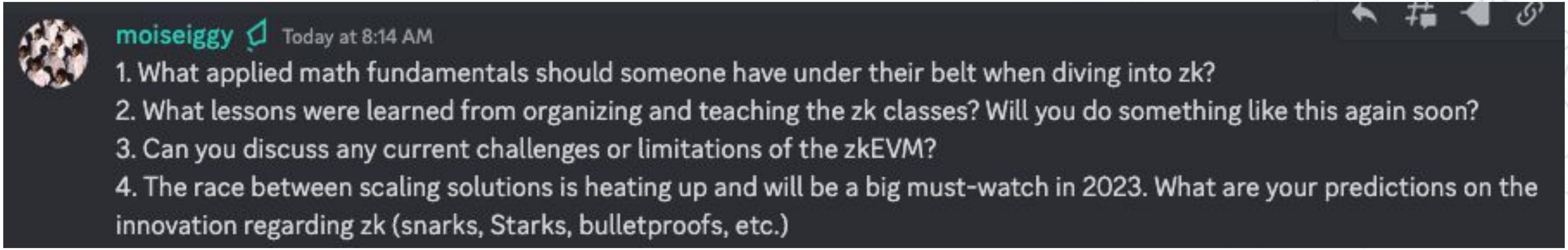
moiseiggy Today at 8:14 AM

1. What applied math fundamentals should someone have under their belt when diving into zk?
2. What lessons were learned from organizing and teaching the zk classes? Will you do something like this again soon?
3. Can you discuss any current challenges or limitations of the zkEVM?
4. The race between scaling solutions is heating up and will be a big must-watch in 2023. What are your predictions on the innovation regarding zk (snarks, Starks, bulletproofs, etc.)

Challenges or limitations of zkEVM

- General purpose circuits are less efficient
- Already maxing out AWS instances to generate proofs
- Fragmentation

Q&A



moiseiggy Today at 8:14 AM

1. What applied math fundamentals should someone have under their belt when diving into zk?
2. What lessons were learned from organizing and teaching the zk classes? Will you do something like this again soon?
3. Can you discuss any current challenges or limitations of the zkEVM?
4. The race between scaling solutions is heating up and will be a big must-watch in 2023. What are your predictions on the innovation regarding zk (snarks, Starks, bulletproofs, etc.)

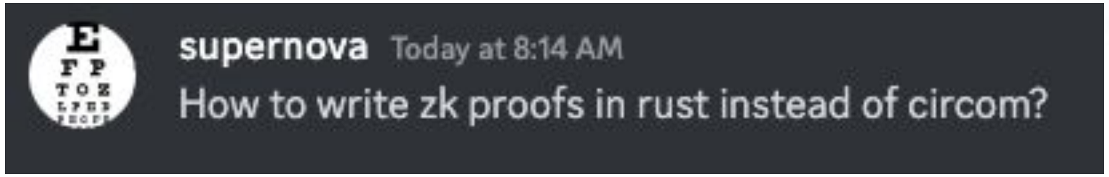
KUDELSKI SECURITY

Predictions for 2023

- ZK programming becomes more common
- Privacy debate
- Few will know the technical differences between scaling solutions. Marketing wins

KUDELSKI SECURITY

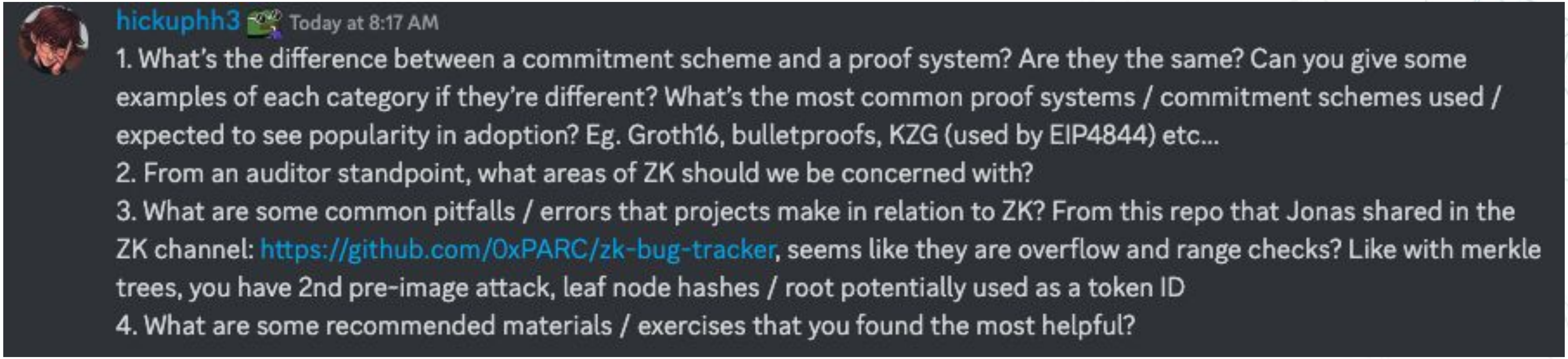
Q&A


supernova Today at 8:14 AM
How to write zk proofs in rust instead of circom?

Arkworks: https://github.com/arkworks-rs

Pratyush Mishra: https://twitter.com/zkproofs
^the best zk teacher

Q&A

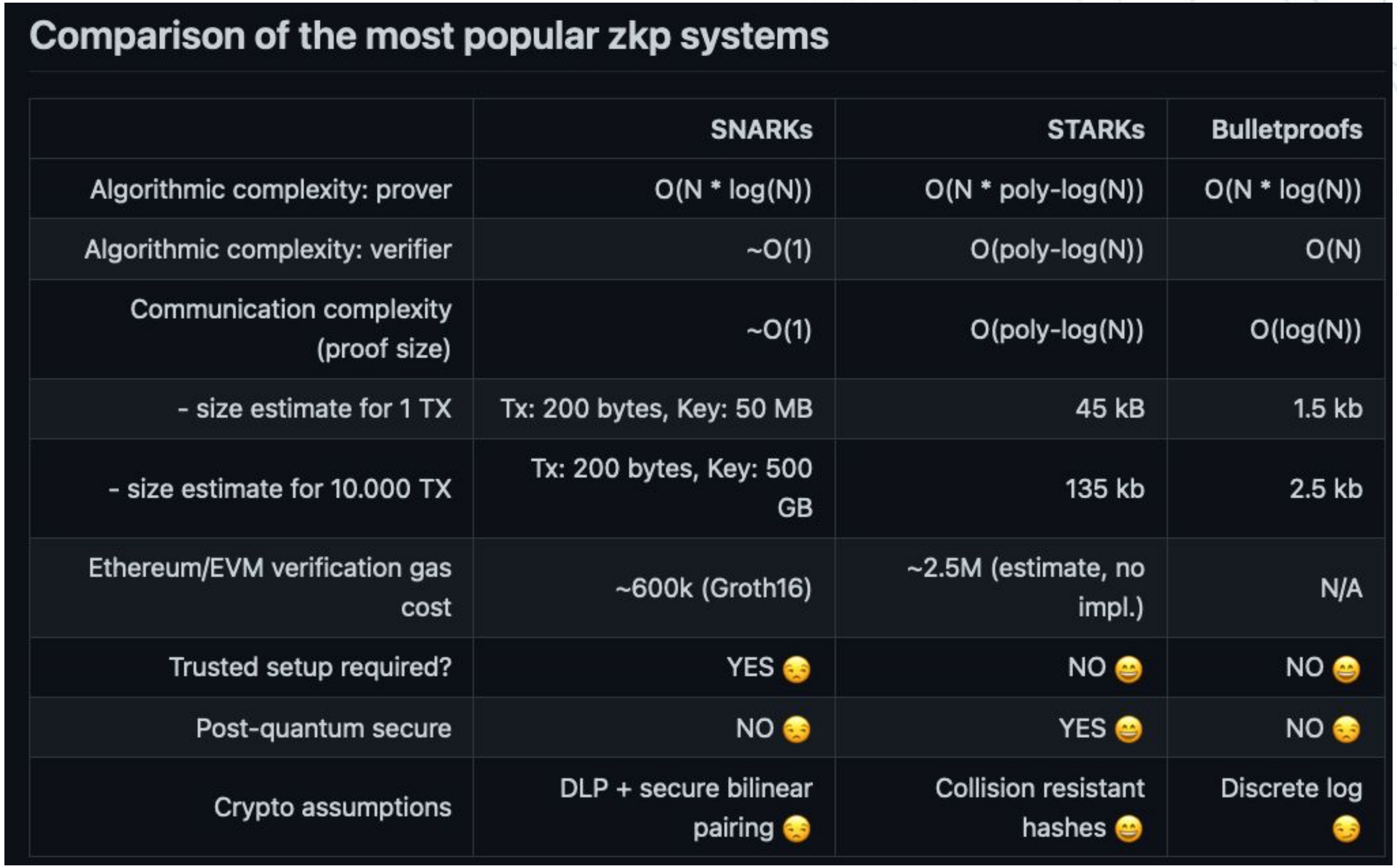


hickuphh3 Today at 8:17 AM

1. What's the difference between a commitment scheme and a proof system? Are they the same? Can you give some examples of each category if they're different? What's the most common proof systems / commitment schemes used / expected to see popularity in adoption? Eg. Groth16, bulletproofs, KZG (used by EIP4844) etc...
2. From an auditor standpoint, what areas of ZK should we be concerned with?
3. What are some common pitfalls / errors that projects make in relation to ZK? From this repo that Jonas shared in the ZK channel: https://github.com/0xPARC/zk-bug-tracker, seems like they are overflow and range checks? Like with merkle trees, you have 2nd pre-image attack, leaf node hashes / root potentially used as a token ID
4. What are some recommended materials / exercises that you found the most helpful?

ZK Differences

- KZG:
- https://scroll.io/blog/kzg



Comparison of the most popular zkp systems

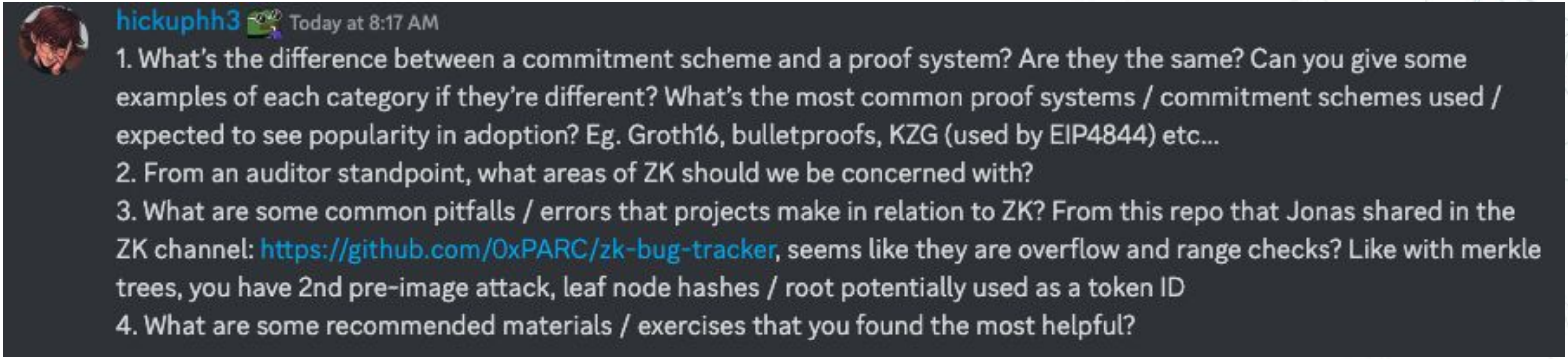
	SNARKs	STARKs	Bulletproofs
Algorithmic complexity: prover	O(N * log(N))	O(N * poly-log(N))	O(N * log(N))
Algorithmic complexity: verifier	~O(1)	O(poly-log(N))	O(N)
Communication complexity (proof size)	~O(1)	O(poly-log(N))	O(log(N))
- size estimate for 1 TX	Tx: 200 bytes, Key: 50 MB	45 kB	1.5 kb
- size estimate for 10.000 TX	Tx: 200 bytes, Key: 500 GB	135 kb	2.5 kb
Ethereum/EVM verification gas cost	~600k (Groth16)	~2.5M (estimate, no impl.)	N/A
Trusted setup required?	YES 😒	NO 😁	NO 😁
Post-quantum secure	NO 😒	YES 😁	NO 😒
Crypto assumptions	DLP + secure bilinear pairing 😒	Collision resistant hashes 😁	Discrete log 😒

ZK Programming

- Circom https://docs.circom.io/getting-started/installation/
- Mina Snarkyjs https://github.com/o1-labs/snarkyjs
- Starkware Cairo https://www.cairo-lang.org/docs/
- Aztec Noir https://aztec.network/noir/
- RISC Zero: https://github.com/risc0/risc0

KUDELSKI SECURITY

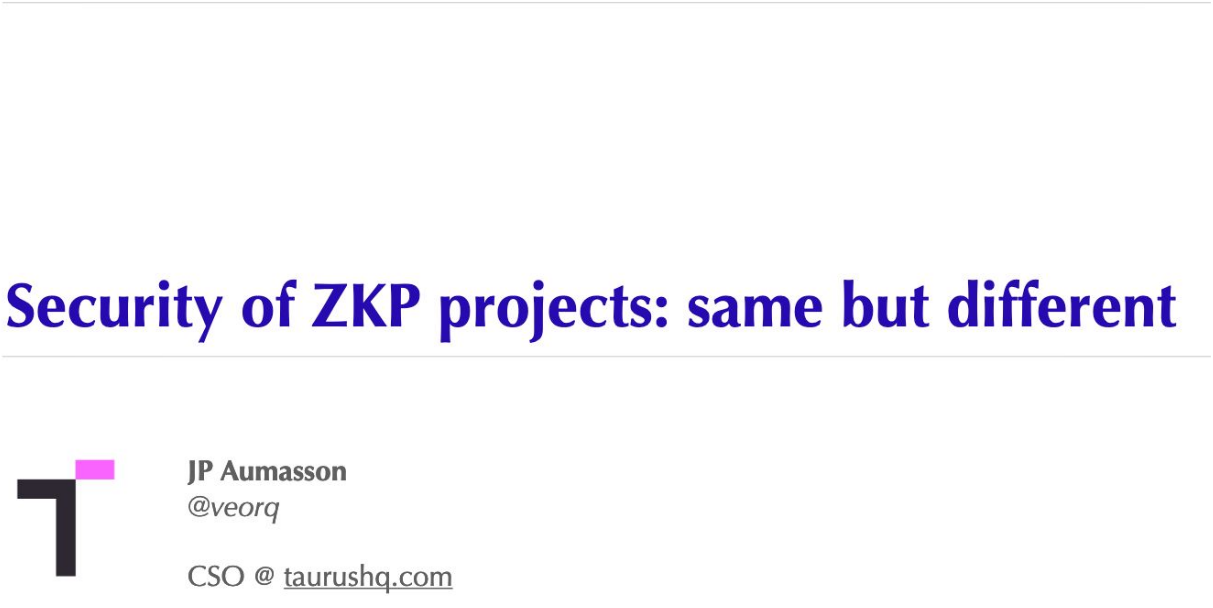
Q&A



hickuphh3 Today at 8:17 AM

1. What's the difference between a commitment scheme and a proof system? Are they the same? Can you give some examples of each category if they're different? What's the most common proof systems / commitment schemes used / expected to see popularity in adoption? Eg. Groth16, bulletproofs, KZG (used by EIP4844) etc...
2. From an auditor standpoint, what areas of ZK should we be concerned with?
3. What are some common pitfalls / errors that projects make in relation to ZK? From this repo that Jonas shared in the ZK channel: https://github.com/0xPARC/zk-bug-tracker, seems like they are overflow and range checks? Like with merkle trees, you have 2nd pre-image attack, leaf node hashes / root potentially used as a token ID
4. What are some recommended materials / exercises that you found the most helpful?

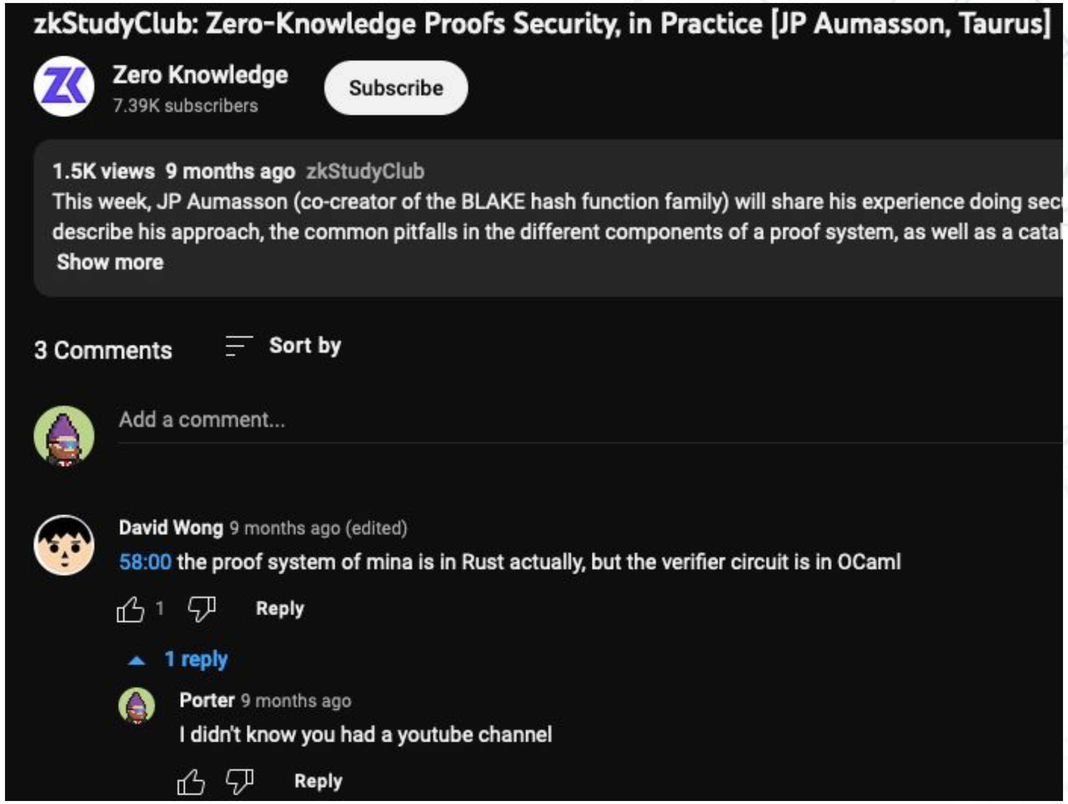
ZK Security:



Security of ZKP projects: same but different

JP Aumasson
@veorq

CSO @ taurushq.com



zkStudyClub: Zero-Knowledge Proofs Security, in Practice [JP Aumasson, Taurus]

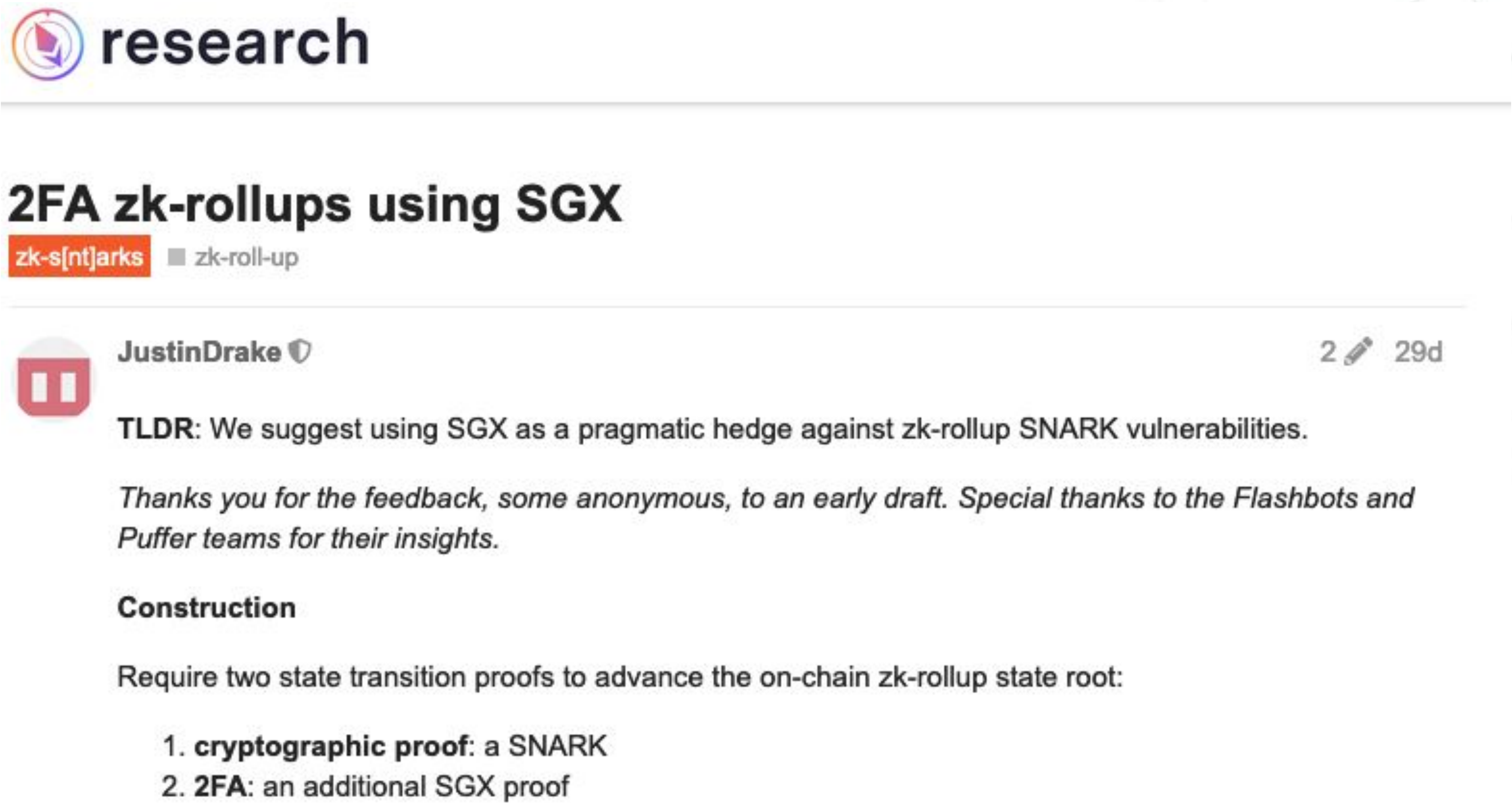
1.5K views 9 months ago zkStudyClub
This week, JP Aumasson (co-creator of the BLAKE hash function family) will share his experience doing sec
describe his approach, the common pitfalls in the different components of a proof system, as well as a catal
Show more

David Wong 9 months ago (edited)
58:00 the proof system of mina is in Rust actually, but the verifier circuit is in OCaml

Porter 9 months ago
I didn't know you had a youtube channel

https://www.youtube.com/watch?v=I_pIrHVz87I

ZK Security: 2FA



research

2FA zk-rollups using SGX

zk-s[nt]arks ■ zk-roll-up

JustinDrake 2 ✎ 29d

TLDR: We suggest using SGX as a pragmatic hedge against zk-rollup SNARK vulnerabilities.

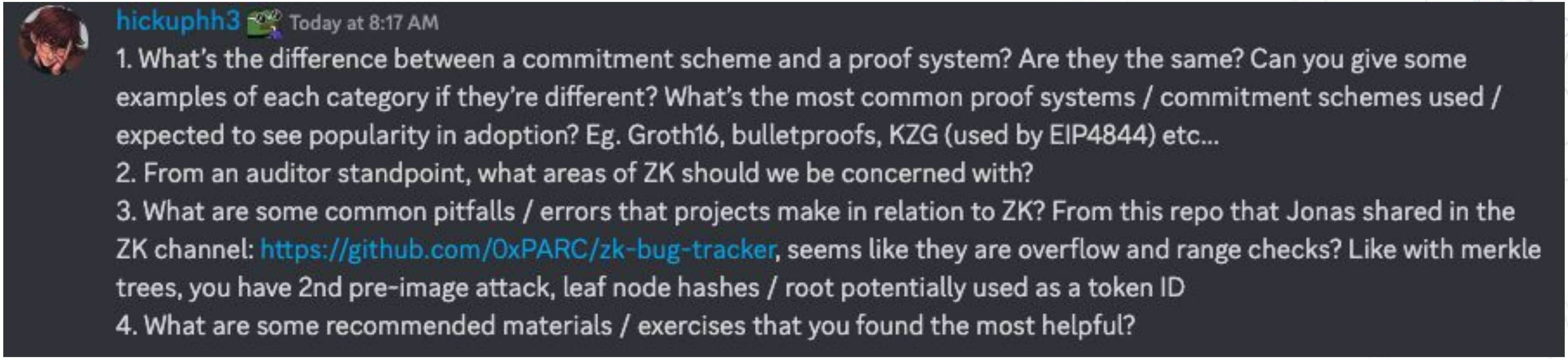
Thanks you for the feedback, some anonymous, to an early draft. Special thanks to the Flashbots and Puffer teams for their insights.

Construction

Require two state transition proofs to advance the on-chain zk-rollup state root:

1. **cryptographic proof**: a SNARK
2. **2FA**: an additional SGX proof

Q&A



hickuphh3 Today at 8:17 AM

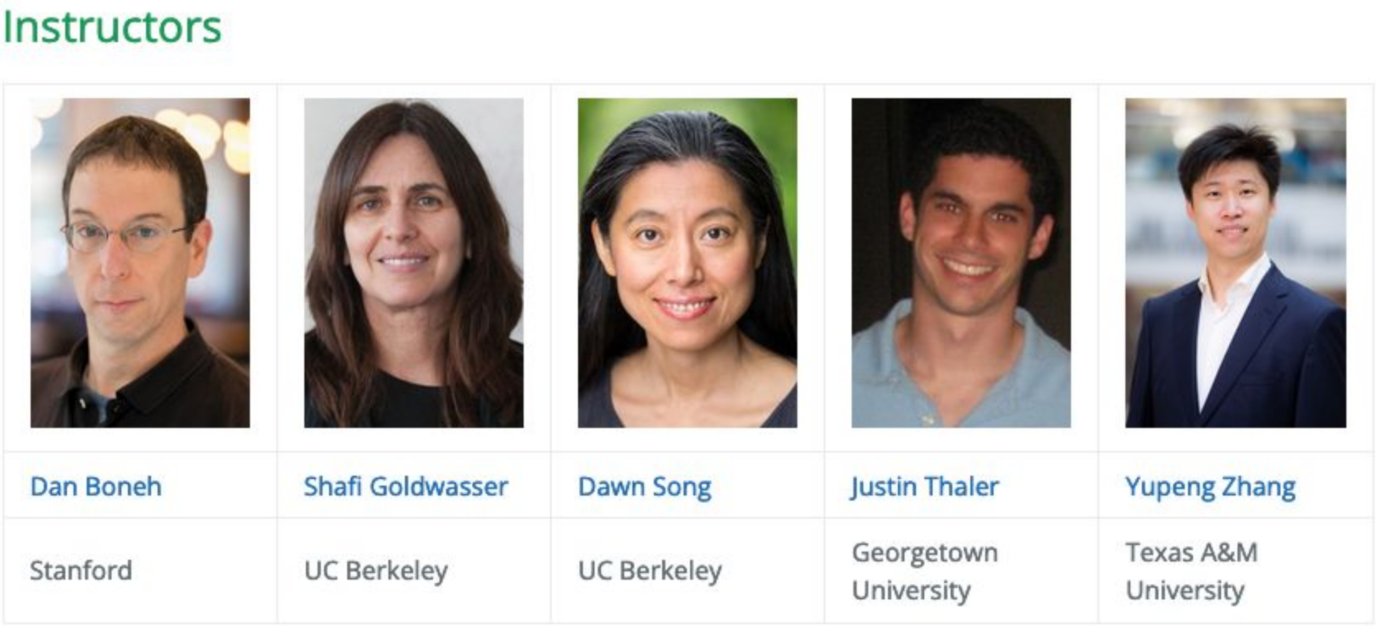
1. What's the difference between a commitment scheme and a proof system? Are they the same? Can you give some examples of each category if they're different? What's the most common proof systems / commitment schemes used / expected to see popularity in adoption? Eg. Groth16, bulletproofs, KZG (used by EIP4844) etc...
2. From an auditor standpoint, what areas of ZK should we be concerned with?
3. What are some common pitfalls / errors that projects make in relation to ZK? From this repo that Jonas shared in the ZK channel: https://github.com/0xPARC/zk-bug-tracker, seems like they are overflow and range checks? Like with merkle trees, you have 2nd pre-image attack, leaf node hashes / root potentially used as a token ID
4. What are some recommended materials / exercises that you found the most helpful?

ZK Study resources

- https://github.com/matter-labs/awesome-zero-knowledge-proofs
- https://scroll.io/blog/
- https://zk-learning.org/
- https://zeroknowledge.fm/

People to follow

- Pratyush Mishra (Aleo + Arkworks): https://twitter.com/zkproofs
- Toghrul Maharramov (Scroll): https://twitter.com/toghrulmaharram
- Psuedo Theos (Scroll): https://twitter.com/pseudotheos
- Daira Hopwood (Zcash): https://twitter.com/feministPLT
- Justin Drake (Ethereum): https://twitter.com/drakefjustin



Instructors

Dan Boneh	Shafi Goldwasser	Dawn Song	Justin Thaler	Yupeng Zhang
Stanford	UC Berkeley	UC Berkeley	Georgetown University	Texas A&M University

Q&A

- More questions?

KUDELSKI SECURITY

KUDELSKI SECURITY

Thanks!

Twitter	Cyber_porter
Telegram	portport12