Privacy in Crypto

Proving Facts Without Revealing Secrets

Day 8 of 10

Prof. Jörg Osterrieder

BSc Seminar: Digital Finance

Spring 2026

Prove You're Rich Without Showing Your Bank Statement

The scenario:
Alice wants a DeFi loan. The lender needs to know her assets exceed \$100K. But Alice does not want to reveal:

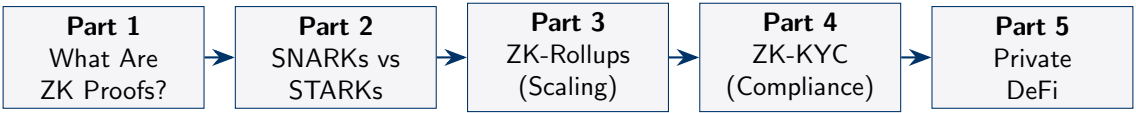
- Which assets she holds
- How much of each
- Her transaction history
- Her identity

With zero-knowledge proofs:
Alice generates a mathematical proof. The lender checks it and learns *exactly one fact*: "assets $>$ \$100K."
Nothing else.

Traditional loan
Submit bank statements, tax returns, passport.
Bank sees **everything**.

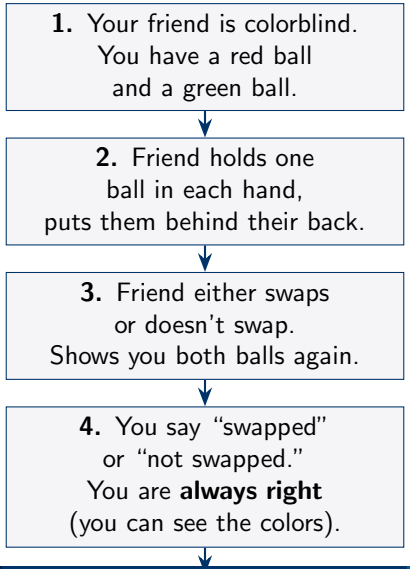
ZK loan
Submit math proof.
Lender sees only:
✓ assets $>$ \$100K

Today's Agenda



Outline

The Colorblind Balls Story



Three Properties of ZK Proofs

Completeness

If the statement is **true**, an honest prover can always convince the verifier.

"If the balls really are different colors, you always get it right."

Soundness

If the statement is **false**, no cheating prover can fool the verifier.

"If both balls are the same color, you cannot consistently guess right."

Zero-Knowledge

The verifier learns **nothing** beyond "the statement is true."

"Your friend learns the balls differ but NOT which is red and which is green."

This is math, not magic. First proven in 1985 by Goldwasser, Micali, and Rackoff [1].

ZK Proof with Sudoku

Claim: "I know a valid solution to this Sudoku puzzle."

1. I randomly relabel all digits ($1 \to 5$, $2 \to 8$, $3 \to 1$, ...). The solution is still valid.
2. I seal each cell in an envelope (cryptographic commitment).
3. **You pick** one row, column, or 3×3 box — your choice.
4. I open those 9 envelopes. You check: all 9 values different? ✓

Why this works:

- If my solution is valid: every row/column/box has 9 distinct values. You always accept.
- If invalid: at least one row/column/box has duplicates. Each round has $\geq 1/27$ chance of catching me.
- After 100 rounds: probability of cheating $< 2.3\%$.

Why it is zero-knowledge:
The random relabeling means the 9 numbers you see are completely meaningless — they tell you **nothing** about my actual solution.

ZK Proofs in Finance: What Can You Prove?

Statement	What verifier learns	What stays hidden
"My assets $>$ \$100K"	Balance is sufficient	Which assets, how much
"I am over 18"	Age requirement met	Exact age, birthdate
"I am not sanctioned"	Compliance check passed	Name, nationality
"I own this wallet"	Account ownership	Private key
"This batch of 2K txs is valid"	All transactions correct	Individual details

The power of ZK: prove *exactly what is needed* and *nothing more*.

Outline

Two Flavors of ZK: SNARKs and STARKs

Property	ZK-SNARK	ZK-STARK
Proof size	**288 bytes** (tiny)	∼45 KB (larger)
Verification time	**3 ms** (fast)	∼50 ms (slower)
Trusted setup	Yes (ceremony needed)	**No** (transparent)
Quantum-safe	No	**Yes**
On-chain cost	∼\$0.50	∼\$5.00
Used by	Zcash, zkSync	StarkNet, dYdX

SNARK = **S**uccinct **N**on-interactive **AR**gument of **K**nowledge
STARK = **S**calable **T**ransparent **AR**gument of **K**nowledge

SNARKs vs. STARKs: The Analogy

SNARK = Notarized Document

- Compact and fast to check
- But you **trust the notary**
- If the notary is corrupt, they can forge fake proofs
- The "trusted setup ceremony" is the notary

STARK = Mathematical Derivation

- Anyone can verify from scratch
- **No trust needed**
- But the derivation is longer (bigger proof)
- Safe even against future quantum computers

Which is better? Depends on the application.

- Need tiny proofs + cheap on-chain verification? $\rightarrow$ **SNARK**
- Need no trust + quantum resistance? $\rightarrow$ **STARK**

What Is This "Trusted Setup"?

The "Powers of Tau" ceremony:

1. Thousands of participants each contribute random data
2. Random data is combined to create the system parameters
3. Each participant **destroys** their randomness
4. If **at least ONE** participant is honest, the system is secure

Ethereum's ceremony had **80,000+** participants.

The "Toxic Waste" Problem

If ALL 80,000 participants secretly colluded and kept their randomness, they could **forge fake proofs**.

This is essentially impossible — but it is a trust assumption that STARKs avoid entirely.

Outline

Ethereum's Bottleneck

Ethereum today:

- $\sim$15 transactions per second
- $\sim$\$5 per transaction
- 10,000 transactions take 11 min
- Total gas cost: \$50,000

Not enough for a global financial system.

The ZK-rollup solution:

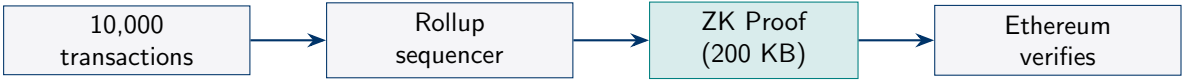
- $\sim$2,000 transactions per second
- $\sim$\$0.05–0.10 per transaction
- 10,000 transactions take 2 min
- Total gas cost: \$500

50$\times$ cheaper. 100$\times$ more throughput.

ZK-Rollups: How They Work

The Teacher Analogy

Instead of the teacher grading every homework problem individually, the students batch 2,000 problems and submit **one proof** that every answer is correct. The teacher checks this single proof in seconds.

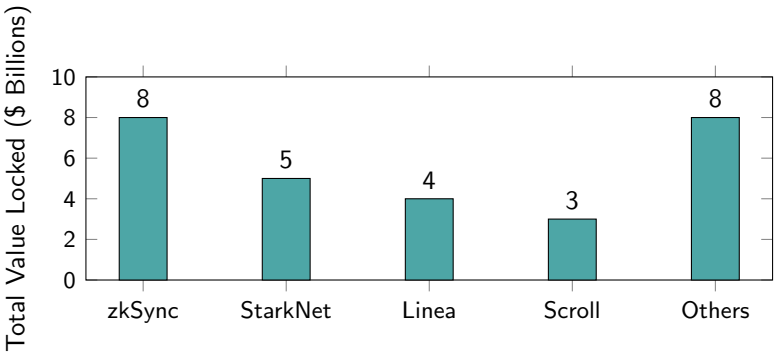


$50,000 if done on L1

$500 total

1 verification

The ZK-Rollup Landscape (2025)



Total ZK-rollup TVL: $\sim$**$28 billion** (2025).
These are real projects handling billions in daily transactions.

Outline

The Impossible Problem (That ZK Solves)

The Regulator Says

"You MUST verify who your users are. Know Your Customer (KYC) is the law. No anonymous trading."

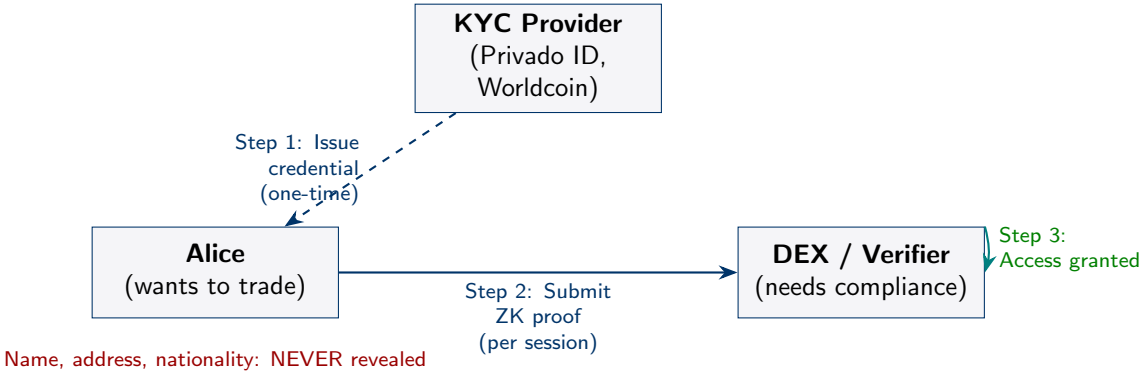
The Privacy Advocate Says

"My identity is MY business. I should be able to trade without being surveilled."

ZK-KYC Says: Both Are Right

Prove you are **compliant** without revealing your **identity**.
The verifier learns "this person is cleared" — but NOT "this person is Alice from Paris."

ZK-KYC: How It Works



What Each Party Knows

Information	KYC Provider	DEX	Blockchain
Alice's name	✓	✗	✗
Alice's nationality	✓	✗	✗
Alice's exact age	✓	✗	✗
"Over 18"	✓	✓	✗
"Not sanctioned"	✓	✓	✗
"EU resident"	✓	✓	✗
"KYC passed"	✓	✓	✓
Trading details	✗	✓	✓

No personal data ever touches the blockchain.
The DEX knows "someone passed KYC" but not *who*.

Outline

DeFi Today: A Glass House

What anyone can see on Etherscan:

- Your wallet address
- Every transaction you ever made
- Your current balances
- Who you traded with
- How much you paid in fees

All of it. Forever.
Your financial life is an open book.

Why this matters:

- Competitors see your trading strategy
- Employers can check your finances
- Stalkers can track your wealth
- Front-runners exploit your trades
- No concept of "financial privacy"

Would you be OK if everyone could see your bank account?

Transparent vs. Private DeFi

Transparent (Today)

From: 0xAlice...
To: Uniswap Router
Input: 10.0 ETH
Output: 30,000 USDC
Gas: 0.005 ETH

Everyone sees everything.

Private (Aztec Network)

From: ENCRYPTED
To: ENCRYPTED
Input: ENCRYPTED
Output: ENCRYPTED
Proof: VALID ✓

Provably valid. Completely private.

Selective disclosure: If a regulator demands Alice's records, she provides a "viewing key" that decrypts *only her* transactions.

Outline

Case Study: Tornado Cash

What it was:

- Privacy mixer on Ethereum
- Deposit ETH, withdraw from different address
- Breaks the link between sender and receiver
- $7.6 billion in total deposits (2019–2022)

What happened:

- Used by privacy-seeking users (majority)
- Also used by North Korean hackers (minority)
- US Treasury **sanctioned** the smart contract (Aug 2022)
- Developer sentenced to 64 months in prison (2024)

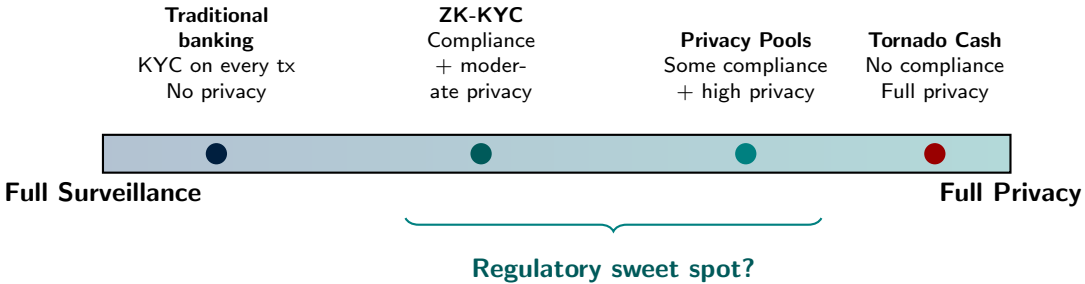
The Legal Question

Is publishing open-source code a crime?

Defense: "A knife maker is not guilty of murder."

Prosecution: "They knew it was used for money laundering and added no compliance."

The Privacy–Compliance Spectrum



ZK technology makes the middle ground possible. Prove compliance without surveillance.

Outline

Exercise Part 1: Hash Function Demo

What Is a Hash?

A hash function takes any input and produces a fixed-size "fingerprint." You cannot reverse it — knowing the fingerprint does not reveal the input.

Try it: Go to `sha256.online` and hash these:

1. Your name → a unique 64-character string
2. Change one letter → **completely different** output
3. Someone else's name → different output (no collisions)

Connection to ZK: In the Sudoku proof, each cell was "committed" using a hash. The verifier sees the hash but not the value — until the prover opens it. This is the foundation of ZK proofs.

Exercise Part 2: The ZK Puzzle

The Game

I am thinking of a number between 1 and 100. I will prove to you that my number is **greater than 50** without telling you what it is.

Protocol:

1. I write my number on paper and seal it in an envelope
2. I give you a hash of my number: `sha256(``73_secret_salt'')`
3. I prove: "my number > 50" (I reveal this fact, not the number)
4. You believe me. Later, I open the envelope. You verify the hash matches.

Class exercise: In pairs, play this game. One person picks a number, the other verifies the proof. Discuss: what goes wrong if I do not use a "salt"?

Day 8: Key Takeaways

1. **Zero-knowledge proofs** let you prove facts without revealing secrets — the foundation of private finance
2. Two flavors: **SNARKs** (small + fast but need trust) vs. **STARKs** (no trust needed but bigger proofs)
3. **ZK-rollups** scale Ethereum $\sim 100\times$ by batching thousands of transactions into one proof
4. **ZK-KYC** resolves the privacy–compliance tension: prove compliance without revealing identity
5. **Private DeFi** (Aztec) makes transactions verifiable but invisible — with selective disclosure for regulators

Discussion Questions

1. Should DeFi be private by default or transparent by default?
2. Was the Tornado Cash developer prosecution fair? Is code speech?
3. Would you pay a premium for privacy in your financial transactions?
4. Where on the privacy–compliance spectrum should regulators draw the line?

Next Time: Prediction Markets

Day 9: When Crowds Beat Experts

- How prediction markets work (price = probability)
- The automated market maker (LMSR)
- Polymarket and the 2024 election
- When markets beat polls — and when they do not
- Manipulation, wash trading, and market limits

References I

[1] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. “The Knowledge Complexity of Interactive Proof Systems”. In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208.