Privacy, Zero-Knowledge Proofs, and Scaling

Day 8 of 10

Prof. Jörg Osterrieder

MSc Seminar: Digital Finance

Spring 2026

MSc Seminar: Digital Finance

Week 2: Advanced Topics in Digital Finance

Today's mission: How to prove facts without revealing secrets, and how that enables both scaling and privacy in digital finance.

- Zero-knowledge proofs: completeness, soundness, zero-knowledge
- SNARKs vs. STARKs: the engineering trade-offs
- ZK-rollups: scaling Ethereum 100×
- ZK-KYC: compliance *without* surveillance
- Private DeFi: the Aztec vision

Proving You Are Rich Without Showing Your Bank Statement

The Scenario

Alice wants a DeFi loan. She must prove her total assets exceed \$100K. In traditional finance, she submits bank statements, tax returns, passport. The lender sees **everything**.

Traditional loan application:

- Bank statements (3 months)
- Tax returns
- Passport / ID
- Employer verification
- *Lender sees all your data*

ZK-powered loan application:

- Alice generates a cryptographic proof
- Proof says: "assets $>$ \$100K"
- Lender verifies the proof: **VALID**
- Lender learns *nothing else*
- No personal data revealed

The Transparency Problem of Current Blockchains

Everything on Ethereum is public. When you transact on-chain:

Data Point	Visible To
Your wallet address	Everyone, forever
Every transaction you have ever made	Everyone, forever
Your token balances	Everyone, forever
Which DeFi protocols you use	Everyone, forever
Your counterparties	Everyone, forever

Consequence

Your entire financial life is a public record. Employers, advertisers, and adversaries can profile you from your wallet address.

Why Privacy Matters for Institutional Finance

Institutions will not adopt DeFi if competitors can see their positions.

- A hedge fund buys 10,000 ETH on-chain $\Rightarrow$ everyone sees the trade and front-runs
- A bank provides liquidity $\Rightarrow$ competitors analyze their strategy in real time
- A corporation pays salaries on-chain $\Rightarrow$ compensation data is public
- An individual swaps tokens $\Rightarrow$ their entire wealth becomes traceable

The question: Can we have blockchain's transparency benefits (auditability, trustlessness) *without* exposing private data?

Answer: Yes. Zero-knowledge proofs make this possible.

Today: ZK Proofs, Scaling, and Privacy

1. Zero-Knowledge Proof Fundamentals
2. SNARKs vs. STARKs
3. ZK-Rollups: Scaling with Proofs
4. ZK-KYC: Privacy-Preserving Compliance
5. Private DeFi: Aztec and Beyond
6. Hands-On: ZK Concepts and Privacy Debate

What Is a Zero-Knowledge Proof?

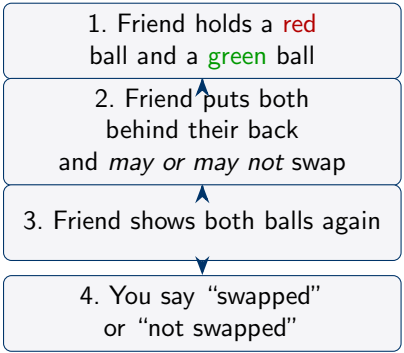
Definition 1 (Zero-Knowledge Proof [ ])

A ZKP is a cryptographic protocol where a **Prover** convinces a **Verifier** that a statement x is true, without revealing any information beyond the truth of x.

Three fundamental properties:

1. **Completeness:** If the statement is true, an honest prover can always convince an honest verifier.
2. **Soundness:** If the statement is false, no cheating prover can convince the verifier (except with negligible probability ε).
3. **Zero-knowledge:** The verifier learns *nothing* beyond the single bit "true." Formally: $\exists$ simulator S such that $\text{View}_V[P(w) \leftrightarrow V(x)]$ is computationally indistinguishable from $S(x)$.

Intuition: The Colorblind Friend Experiment



Analysis:

If balls are truly different colors:

- You can *always* answer correctly
- After 20 rounds: $P(\text{lucky guess}) = (1/2)^{20}$
  $= 9.5 \times 10^{-7}$ (negligible)

Completeness: ✓ (you always get it right)

Soundness: ✓ (cheater caught in $\sim$20 rounds)

Zero-knowledge: ✓ (friend still does not know *which* is red and *which* is green)

Worked Example: Sudoku Zero-Knowledge Proof

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

1. Prover has a completed grid (the *witness*).
2. Prover randomly permutes digits 1–9 (e.g., $1 \to 5$, $2 \to 8$, ...). This relabeling preserves validity.
3. Prover commits to each cell (hash of value + random nonce).
4. Verifier randomly picks one row, column, or 3×3 box (27 choices).
5. Prover opens those 9 commitments.
6. Verifier checks: are all 9 values *distinct*?

Soundness after k rounds:

$$P(\text{cheating undetected}) = \left(\frac{26}{27}\right)^k \xrightarrow{k=100} 0.023 \xrightarrow{k=500} < 10^{-8}$$

Zero-knowledge: The random permutation ensures the verifier learns nothing about the actual solution.

Interactive vs. Non-Interactive Proofs

Interactive ZKP:

- Multi-round protocol (Prover $\leftrightarrow$ Verifier)
- Verifier sends random challenges
- Prover responds to each challenge
- Requires both parties to be online
- Example: the colorblind ball experiment

Non-Interactive ZKP (NIZKP):

- Single message from Prover to Verifier
- **Fiat–Shamir heuristic**: replace verifier's random challenge with hash of the transcript
- Proof can be verified by anyone, anytime
- Suitable for blockchain (post proof on-chain)

Key insight: Blockchain verification requires non-interactive proofs. The verifier is a smart contract that checks a proof in a single transaction—no back-and-forth.

What Can Be Proved with Zero Knowledge?

Any statement in the complexity class NP can be proved in zero knowledge.

Finance Application	**ZK Statement**
Solvency proof	"My assets $>$ liabilities"
Credit check	"My credit score > 700"
KYC compliance	"I am not on the sanctions list"
Age verification	"I am over 18"
Tax compliance	"I paid taxes on this income"
Transaction validity	"This transfer follows the protocol rules"
Portfolio constraint	"My leverage ratio $< 3\times$"

In each case, the verifier learns *only* that the statement is true—not the underlying private data.

Checkpoint: ZKP Properties

Quick Question

Alice claims she knows the password to a system but does not want to reveal it. She runs a ZKP protocol and the verifier accepts after 30 rounds. Which property guarantees that Alice is not bluffing?

Checkpoint: ZKP Properties

Quick Question

Alice claims she knows the password to a system but does not want to reveal it. She runs a ZKP protocol and the verifier accepts after 30 rounds. Which property guarantees that Alice is not bluffing?

Answer

Soundness. After 30 rounds, the probability that a cheating prover fools the verifier is at most $(1/2)^{30} \approx 10^{-9}$. Soundness guarantees that false statements are rejected with overwhelming probability.

ZK-SNARKs: Succinct Non-Interactive Arguments of Knowledge

ZK-SNARK = **S**uccinct **N**on-interactive **AR**gument of **K**nowledge

- **Succinct:** Proof is tiny ($\sim$288 bytes for Groth16) and fast to verify ($\sim$3 ms)
- **Non-interactive:** Single message from prover to verifier
- **Argument of knowledge:** Prover must "know" a witness, not just that one exists

The catch: trusted setup.

- Requires a ceremony ("powers of tau") to generate public parameters
- N participants each contribute randomness
- If *at least one* participant is honest (destroys their secret), the setup is secure
- If *all* collude: they can forge proofs ("toxic waste")
- Ethereum's ceremony: 80,000+ participants

ZK-STARKs: Scalable Transparent Arguments of Knowledge

ZK-STARK = **S**calable **T**ransparent **AR**gument of **K**nowledge

- **Scalable:** Prover time is quasi-linear in circuit size
- **Transparent:** *No trusted setup*—all randomness is public
- Based on hash functions (collision resistance), not elliptic curves
- **Post-quantum secure:** not broken by quantum computers

The trade-off: Proofs are much larger.

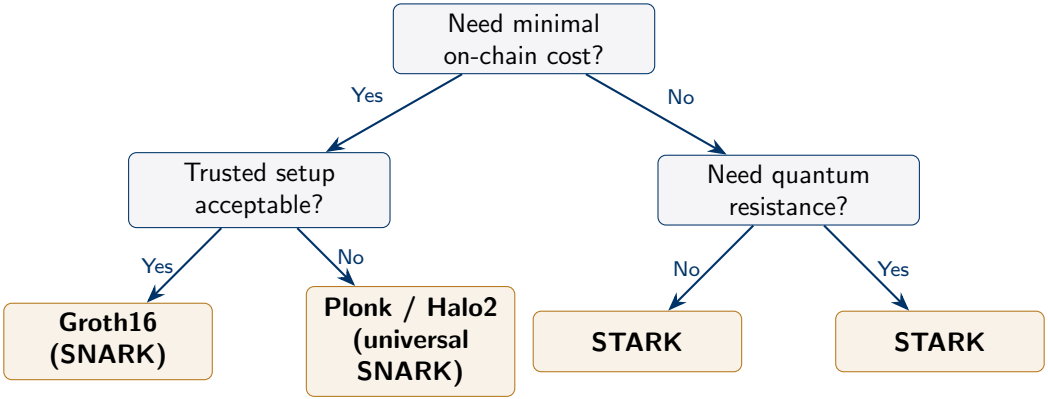
- STARK proof: ∼45–200 KB (vs. 288 bytes for Groth16 SNARK)
- Verification: ∼50 ms (vs. ∼3 ms for SNARK)
- On-chain cost: ∼2M gas (∼\$5) vs. ∼200K gas (∼\$0.50)

SNARKs vs. STARKs: Head-to-Head [ ]

Property	ZK-SNARK (Groth16)	ZK-STARK
Proof size	288 bytes	$\sim$45 KB
Verification time	$\sim$3 ms	$\sim$50 ms
Prover time (1M gates)	$\sim$30 s	$\sim$2 min
Trusted setup	Yes	No
Post-quantum secure	No	Yes
Gas cost on Ethereum	$\sim$200K ($\sim$\$0.50)	$\sim$2M ($\sim$\$5.00)
Cryptographic basis	Elliptic curves (ECDL)	Hash functions
Used by	Zcash, zkSync, Tornado Cash	StarkNet, StarkEx, dYdX (v4 settlement)

Rule of thumb: SNARKs when gas cost matters (cheap on-chain verification); STARKs when trust assumptions matter (no ceremony, quantum-safe).

Choosing a Proof System: Decision Tree



Plonk (2019): a "universal SNARK" requiring only one trusted setup for all circuits—a middle ground between Groth16 and STARKs.

ZK Proof Systems in Production (2025)

Protocol	Proof System	TVL	Use Case
zkSync Era	Plonk (SNARK)	\$8B	General-purpose ZK-rollup
StarkNet	STARK	\$5B	General-purpose ZK-rollup
Linea	SNARK	\$4B	Ethereum L2
Scroll	SNARK (KZG)	\$3B	EVM-equivalent rollup
Polygon zkEVM	SNARK	\$2B	EVM-compatible rollup
Aztec	SNARK (Noir)	\$0.5B	Private smart contracts
Total ZK-rollup TVL		**\$28B+**	

ZK technology has moved from academic curiosity to \$28B+ of real value secured. The "ZK winter" is over.

The Blockchain Scalability Trilemma

Ethereum L1 limitations:

- Throughput: $\sim$15 TPS
- Cost: $\sim$\$5 per transfer, \$20+ for swaps
- 10,000 transactions: 11 minutes, \$50,000 in gas

Traditional scaling approaches:

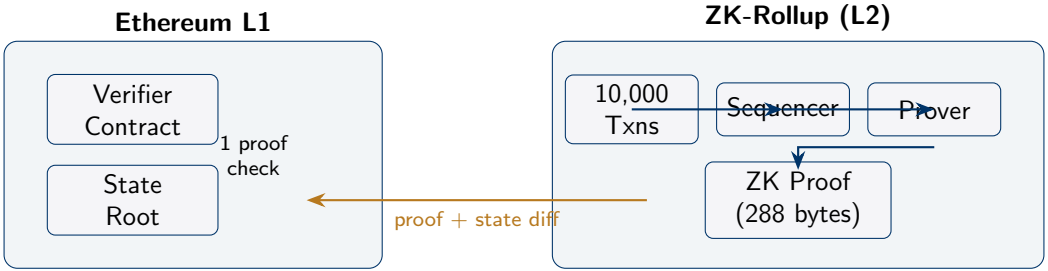
- Bigger blocks: sacrifices decentralization
- Sidechains: weaker security guarantees
- State channels: limited to specific use cases

ZK-rollup approach:

- Process transactions off-chain
- Generate a ZK proof of validity
- Post *only the proof* to L1
- Ethereum verifies the proof, not the transactions
- Result: 10–100$\times$ throughput, inherited L1 security

Analogy: Instead of a teacher grading every homework problem, the ZK-rollup submits a proof that *all* problems are correct. One verification replaces thousands.

ZK-Rollup Architecture: L1 vs. L2



	L1 Only	ZK-Rollup
Throughput	15 TPS	2,000 TPS
Cost per tx	\$5.00	\$0.05–\$0.10
Finality	12 min	∼2 min + L1 finality
Security	L1 native	Inherits L1

ZK-Rollup Cost Savings: Worked Example

Batch of 10,000 token transfers:

Component	Ethereum L1	zkSync Era (L2)
Per-tx verification	21,000 gas $\times$ 10K	—
ZK proof verification	—	200,000 gas (once)
Compressed state diff	—	$\sim$300,000 gas
Total gas	210,000,000	500,000
Total cost (at 30 gwei)	\$50,000	\$500
Cost per tx	\$5.00	\$0.05
Savings	—	**100**$\times$

Where do the savings come from?

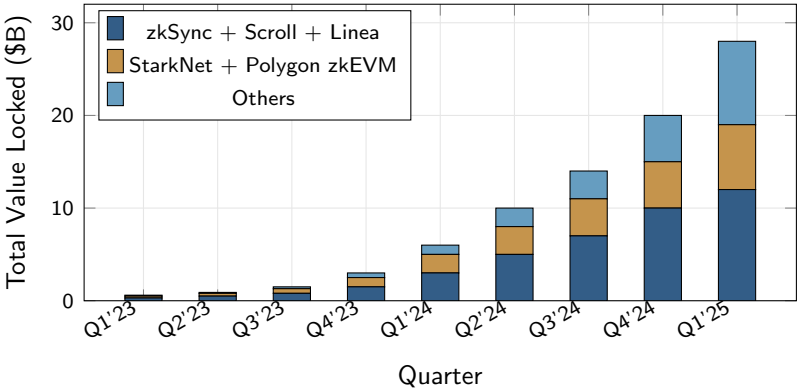
- Ethereum verifies *one proof* instead of 10,000 transactions
- State diffs are compressed ($\sim$20 bytes per tx vs. full calldata)
- Prover computation happens off-chain (cheap commodity hardware)

ZK-Rollups vs. Optimistic Rollups

Property	ZK-Rollup	Optimistic Rollup
Validity mechanism	Cryptographic proof	Fraud proof (dispute period)
Finality on L1	Minutes	7 days (challenge window)
Withdrawal time	Minutes	7 days (or use bridge)
Computation cost	High (proof generation)	Low (just execute)
On-chain cost	Proof verification	State root + calldata
Security assumption	Math (proof soundness)	At least 1 honest verifier
EVM compatibility	Improving (zkEVM)	Full (EVM equivalent)
Examples	zkSync, StarkNet, Scroll	Arbitrum, Optimism, Base

Trend: ZK-rollups are winning on finality speed. Optimistic rollups have the EVM compatibility advantage—but zkEVMs are closing the gap rapidly.

ZK-Rollup TVL Growth



From <$1B in early 2023 to $28B+ in Q1 2025. ZK-rollups are the fastest-growing segment of Ethereum's scaling ecosystem.

The Compliance Dilemma

Regulator says:

- "Know your customer" (KYC)
- "Anti-money laundering" (AML)
- "Sanctions screening"
- Requires personal data collection

$\Rightarrow$ Privacy is the *cost* of compliance.

Privacy advocate says:

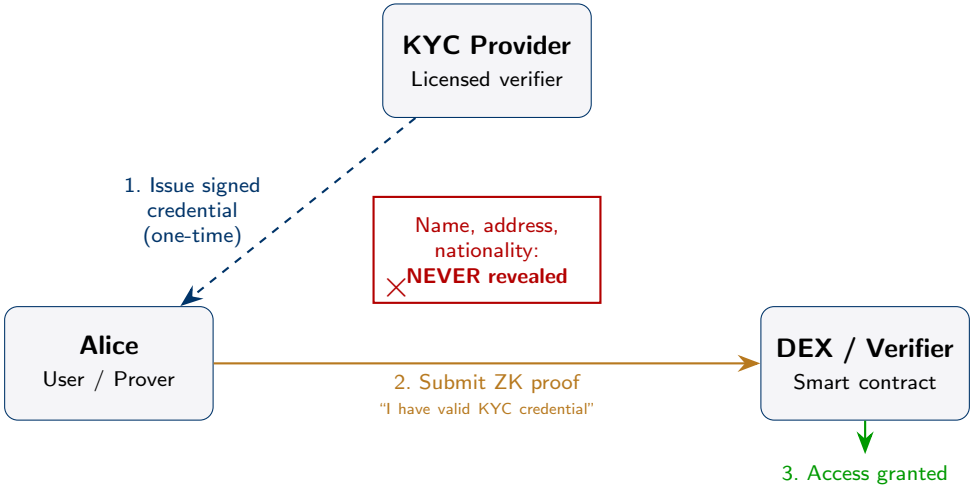
- "My identity is my right"
- "Data breaches expose millions"
- "Surveillance is not security"
- Opposes data collection

$\Rightarrow$ Compliance is the *cost* of privacy.

ZK-KYC resolves this tension

Prove compliance *without* revealing identity. The verifier learns "this person is cleared" but *not* "this person is Alice from Paris."

ZK-KYC: The Three-Party Protocol



ZK-KYC Walkthrough: Alice Trades on a MiCA-Compliant DEX

Step 1: Credential Issuance (off-chain, one-time)

- Alice submits passport + proof of residence to a licensed KYC provider
- Provider issues a signed *Verifiable Credential* (VC): EU resident, AML cleared, not sanctioned, over 18

Step 2: Proof Generation (on Alice's device, $\sim$2 seconds)

- Alice's wallet generates a ZK proof attesting she holds a valid credential
- Proof size: $\sim$300 bytes. Reveals *nothing* about her identity

Step 3: On-Chain Verification

- DEX smart contract verifies: proof valid, issuer trusted, credential not expired
- Alice is granted access. **No PII stored on-chain.**

Privacy guarantee: DEX knows "this wallet passed KYC." DEX does *not* know Alice's name, nationality, age, or which provider issued the credential.

ZK-KYC vs. Traditional KYC: Comparison

Dimension	Traditional KYC	ZK-KYC
Data collected	Full identity	None (proof only)
Data stored by service	Yes (honeypot risk)	No
Data breach risk	High	None
User experience	Days of paperwork	2-second proof
Cross-platform reuse	No (re-KYC each time)	Yes (same credential)
Regulatory compliance	✓	✓
Cost per verification	\$5–\$50	\$0.01 (gas)
Revocation support	Manual	On-chain Merkle tree

With ZK-KYC, every data breach headline becomes "this problem was avoidable." If services never store PII, there is nothing to steal.

Case Study: Tornado Cash and Privacy Pools [ ]

Tornado Cash (2022):

- Privacy mixer: deposit ETH, withdraw from a different address
- \$7.6B total deposits (2019–2022)
- Used by privacy-conscious individuals *and* North Korean hackers
- OFAC sanctioned the smart contract addresses (Aug 2022)
- Developer Alexey Pertsev sentenced to 64 months (2024)

What ZK-KYC could have done:

- Require ZK proof that funds are *not* from sanctioned addresses
- Privacy preserved (no link between deposit and withdrawal)
- Compliance enforced (sanctioned funds excluded)

Privacy Pools (Buterin et al., 2023): users prove their deposit belongs to a set of "known clean" addresses using *association sets*—ZK proof of set membership without revealing the specific address.

Transparent DeFi vs. Private DeFi

Transparent (current Ethereum):

Alice swaps 10 ETH for 30,000 USDC on Uniswap.

Field	Visible?
Sender	0xAlice... ✓
Action	Uniswap swap ✓
Input	10.0 ETH ✓
Output	30,000 USDC ✓
History	All txns ✓

Private (Aztec):

Same swap on an Aztec-based DEX.

Field	Visible?
Sender	ENCRYPTED
Action	ENCRYPTED
Input	ENCRYPTED
Output	ENCRYPTED
Proof	VALID ✓

With selective disclosure: Alice can provide a "viewing key" to a regulator, decrypting *only her* transactions. Everyone else sees nothing.

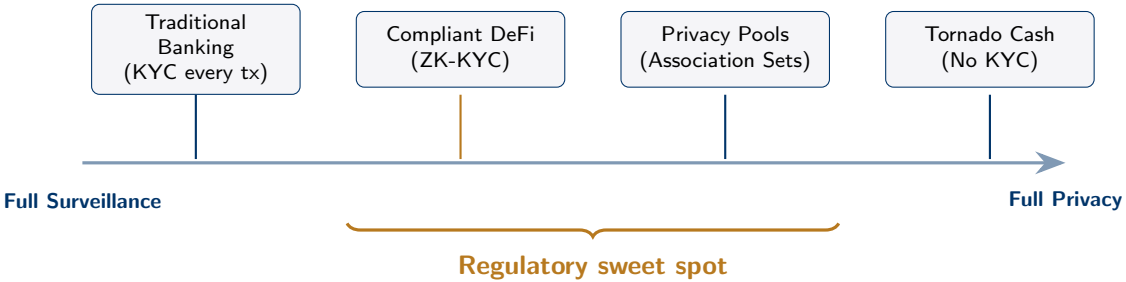
How Aztec Works: Encrypted Notes and Nullifiers

Aztec uses a UTXO model (like Bitcoin, unlike Ethereum's account model):

1. **Notes:** Each balance is an encrypted "note" committed to a Merkle tree. Only the owner (with the decryption key) can read its value.
2. **Spending:** To spend, generate a ZK proof that:
   - The input note commitment is in the Merkle tree (existence)
   - The nullifier has not been used (no double-spend)
   - Output note values are correct (inputs = outputs)
3. **Nullifier:** A unique hash derived from the note, revealed on-chain to prevent double-spending, but unlinkable to the original note.

The privacy guarantee: An observer sees nullifiers and new commitments being created, but cannot link them to users, amounts, or assets.

The Privacy–Compliance Spectrum



The challenge is finding the right balance. ZK-KYC and Privacy Pools offer a middle ground: compliance without surveillance.

The Economics of Privacy in DeFi

Is privacy a luxury good or a public good?

Metric	Transparent DeFi	Private DeFi
MEV extraction	\$500M+/year	Near zero
Front-running risk	High	Eliminated
Institutional adoption	Limited (data leakage)	Enabled
Smart contract cost	Standard	$\sim$3–5$\times$ higher
Proof generation time	N/A	2–30 seconds

Key insight: Privacy eliminates MEV (Maximal Extractable Value) because searchers cannot see pending transactions. The \$500M+ annual MEV extraction is effectively a *tax on transparency*.

Prediction: As ZK proof generation becomes cheaper, private-by-default will become the standard for DeFi.

Checkpoint: Privacy and Compliance

Quick Question

A regulator demands that all DeFi users be identifiable. A ZK-KYC system is proposed where users prove compliance without revealing their identity. The regulator objects: "But we cannot identify criminals."

How would you respond?

Checkpoint: Privacy and Compliance

Quick Question

A regulator demands that all DeFi users be identifiable. A ZK-KYC system is proposed where users prove compliance without revealing their identity. The regulator objects: "But we cannot identify criminals."

How would you respond?

Key Points

1. ZK-KYC prevents *non-compliant* users from accessing the protocol (sanctioned addresses, non-verified users are blocked)
2. Selective disclosure allows regulators with a court order to access specific user records via viewing keys
3. The KYC provider retains the ability to respond to lawful requests
4. This is *more* secure than traditional KYC: no centralized database to breach

Hands-On Session

ZK Concept Exercises and Privacy Debate

Pen-and-paper exercises + structured debate

Exercise 1: ZK Proof Probability Calculation

Setup: You run a ZKP protocol where each round catches a cheater with probability $p = 1/3$.

1. Calculate the probability that a cheater survives k rounds:
   $P(\text{undetected}) = (1-p)^k = (2/3)^k$
2. How many rounds are needed for $P < 10^{-6}$?

$$(2/3)^k < 10^{-6} \implies k > \frac{-6}{\log_{10}(2/3)} = \frac{6}{0.1761} \approx 34.1 \implies k = 35$$

3. If verification costs 0.001 ETH per round, what is the total verification cost for 10^{-6} security?
4. Compare: a SNARK achieves 2^{-128} security in a *single* verification at 200K gas. At what gas price does the SNARK become cheaper?

Exercise 2: SNARK vs. STARK Cost Analysis

A DeFi protocol processes 50,000 transactions per day.

	SNARK (Groth16)	STARK
Gas per proof verification	200,000	2,000,000
Transactions per batch	5,000	5,000
Batches per day	10	10
Gas price (gwei)	30	30

Calculate:

1. Daily gas cost for each proof system (in ETH and USD at \$3,000/ETH)
2. Annual cost difference
3. At what gas price does the cost difference exceed the protocol's annual revenue of \$5M?
4. Qualitative: when is the STARK's higher cost justified?

Exercise 3: ZK-KYC Design Challenge

Design a ZK-KYC system for a tokenized bond platform:

1. **Requirements:**
   - Investors must be accredited (assets $>$ \$1M or income $>$ \$200K)
   - EU investors must comply with MiCA
   - US investors must comply with Reg D
   - Sanctions screening (OFAC, EU sanctions list)
2. **Design questions:**
   - What claims should the Verifiable Credential contain?
   - How do you handle credential expiration and revocation?
   - How do you support multiple jurisdictions with different rules?
   - What is the minimum information the smart contract needs?
3. **Sketch** the credential schema and verification circuit

Debate: Should DeFi Be Private by Default?

Motion

"All DeFi protocols should implement privacy-by-default, with selective disclosure available to regulators upon lawful request."

For the motion:

- Privacy is a fundamental right
- Transparency enables MEV extraction ($500M+/year)
- Institutional adoption requires confidentiality
- Data breaches are eliminated
- ZK-KYC provides sufficient compliance

Against the motion:

- Transparency enables public auditing
- Privacy facilitates money laundering
- Regulators need real-time visibility
- ZK technology is not mature enough
- Tornado Cash showed the risks

Debate: Discussion Starters

Consider these scenarios:

1. A dictator's government uses on-chain surveillance to identify and punish dissidents sending money to opposition groups. Does privacy protect the vulnerable?
2. A terrorist organization uses private DeFi to fund attacks undetected. Does privacy enable harm?
3. A publicly traded company uses transparent DeFi and competitors reverse-engineer their treasury strategy. Is forced transparency anticompetitive?
4. A ZK-KYC system has a bug that allows sanctioned entities to pass verification. Who is liable?

The hard truth: There is no perfect answer. Privacy and compliance exist on a spectrum, and every point on that spectrum has trade-offs.

Key Takeaways: Day 8

1. **Zero-knowledge proofs** let you prove facts without revealing secrets—the foundation of private finance
2. **SNARKs** (small, fast, trusted setup) vs. **STARKs** (bigger, no trust, quantum-safe): know the trade-off for each use case
3. **ZK-rollups** scale Ethereum 100× while inheriting L1 security—\$28B+ TVL and growing
4. **ZK-KYC** resolves the privacy–compliance tension: prove compliance without revealing identity
5. **Private DeFi** (Aztec) makes transactions provably valid but completely invisible—eliminating MEV and enabling institutional adoption

Looking ahead: Tomorrow we explore prediction markets—where market prices become probability estimates and the crowd beats the experts.

Further Reading

Core references:

- Goldwasser, Micali, and Rackoff (1989): foundational ZKP paper [3]
- Ben-Sasson et al. (2018): STARKs [1]
- Buterin et al. (2023): Privacy Pools [2]

Technical deep dives:

- Groth (2016): Groth16 SNARK construction
- Vitalik Buterin (2021): "An Incomplete Guide to Rollups"
- Aztec documentation: `https://docs.aztec.network`

Questions? • Office hours: by appointment • `joerg.osterrieder@usi.ch`

References I

[1] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Scalable, Transparent, and Post-Quantum Secure Computational Integrity". In: *Advances in Cryptology – CRYPTO 2018*. Springer, 2018, pp. 365–393.

[2] Vitalik Buterin et al. *Blockchain Privacy and Regulatory Compliance: Towards a Practical Equilibrium*. Working Paper. 2023.

[3] 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.