Privacy & Zero-Knowledge Proofs

Pre-Class Discovery Handout

Lesson 13 · Complete before class · 25–30 minutes

Activity 1: ZK Proof Intuition Builder — 10 min

Build an intuitive understanding of zero-knowledge proofs before the technical details. Answer:

1. Explain the "Where's Waldo" analogy for zero-knowledge proofs. How does it satisfy the three core properties: **completeness** (an honest prover always convinces a verifier), **soundness** (a cheating prover cannot convince a verifier of a false statement), and **zero-knowledge** (the verifier learns nothing beyond the truth of the statement)?
2. Think of **3 real-world scenarios** where you might want to prove something without revealing the underlying information. For each, state what is *proved* and what *remains hidden* (example: proving you are over 18 without showing your full date of birth or ID number).
3. What is the difference between **interactive** and **non-interactive** zero-knowledge proofs? Why are non-interactive proofs (NIZKs) significantly more useful for blockchain applications where asynchronous verification is required?
4. What is a **"trusted setup"**? Why do some proof systems (like Groth16, used in Zcash) require one, and why is this controversial from a security and trust perspective? What is a "toxic waste" ceremony?

Activity 2: Proof Systems Comparison — 10 min

Research the main families of ZK proof systems and compare their engineering trade-offs. Answer:

1. Look up **SNARKs vs. STARKs**. Compare them across: proof size, verification time, trusted setup requirement, and quantum resistance. Fill in the comparison table below.
2. What does the "S" in SNARK stand for? What does "succinct" mean in concrete terms for blockchain scalability — specifically for on-chain verification costs (gas)?
3. What is the **FRI protocol** (Fast Reed–Solomon Interactive Oracle Proof of Proximity) used in STARKs? In one sentence, why does its algebraic structure avoid the need for a trusted setup?
4. Look up **Bulletproofs**, the proof system used in Monero for confidential transactions. How do they differ from SNARKs in terms of proof generation and verification complexity? What is their main advantage and disadvantage compared to SNARKs?

System	Proof Size	Verify Time	Trusted Setup	Quantum Safe	Used By
Groth16 (SNARK)					
PLONK (SNARK)					
STARK					
Bulletproofs					

Activity 3: Privacy Coins Explorer

5 min

Investigate the two leading privacy-preserving cryptocurrencies and the regulatory environment around them. Answer:

1. Compare **Monero (XMR)** and **Zcash (ZEC)**. What specific cryptographic privacy techniques does each use? Is privacy mandatory (always-on) or optional (shielded vs. transparent) for each?
2. What are **ring signatures**? How does Monero use them to hide the identity of the sender among a group of possible signers, without requiring cooperation from the other group members?
3. What are **stealth addresses**? How does Monero's one-time address scheme protect the receiver by ensuring each transaction generates a fresh, unlinkable destination address?
4. Why have major exchanges (e.g., Kraken UK, Bittrex, Binance in some regions) delisted privacy coins? What is the fundamental regulatory tension between financial privacy and Anti-Money Laundering (AML) / Know Your Customer (KYC) compliance obligations?

Feature	Monero (XMR)	Zcash (ZEC)	Bitcoin
Sender Privacy Technique	_____	_____	_____
Receiver Privacy Technique	_____	_____	_____
Amount Hiding	_____	_____	_____
Privacy Mandatory?	_____	_____	_____
Exchange Listing Status	_____	_____	_____

Activity 4: ZK Applications Discovery

5 min

Explore how zero-knowledge proofs are being deployed in production blockchain systems today. Answer:

1. What is a **ZK-rollup**? How does it use ZK proofs to batch thousands of transactions and post only a succinct validity proof to Ethereum, reducing costs? Name **2 live ZK-rollup projects** and look up their current Total Value Locked (TVL) on **L2Beat.com** (`https://l2beat.com`).
2. What is a **zkEVM**? Why is achieving full EVM-equivalence with ZK proofs considered a major engineering challenge? Name one zkEVM project and identify where it sits on Vitalik Buterin's zkEVM "Type" classification (Type 1–4).
3. How could ZK proofs enable **private voting on a blockchain**? List the properties such a system would require: (a) vote secrecy, (b) eligibility verification, (c) double-vote prevention, and (d) public verifiability of the tally — without a trusted tallier.
4. What is **zkML** (zero-knowledge machine learning)? Why might proving that an AI model was correctly executed on specific inputs be important for financial auditing, AI safety, or regulatory compliance — without revealing the model weights or private input data?

ZK-Rollup Project	TVL (from L2Beat)	Proof System Used	EVM Compatible?
_____	_____	_____	_____
_____	_____	_____	_____

Key Terms

Term	Definition
Zero-Knowledge Proof	A cryptographic protocol in which a prover convinces a verifier that a statement is true without revealing any information beyond the truth of the statement itself. Requires completeness, soundness, and the zero-knowledge property.
Completeness	The property that an honest prover with a valid witness can always convince an honest verifier; i.e., correct proofs are always accepted.
Soundness	The property that a cheating prover cannot convince a verifier of a false statement, except with negligible (cryptographically small) probability.
SNARK	Succinct Non-interactive ARgument of Knowledge. A ZK proof system producing extremely short proofs (e.g., 200 bytes) that can be verified in milliseconds. Typically requires a trusted setup. Examples: Groth16, PLONK.
STARK	Scalable Transparent ARgument of Knowledge. A ZK proof system based on hash functions and the FRI protocol. No trusted setup, post-quantum secure, but produces larger proofs than SNARKs.
Trusted Setup	A one-time ceremony that generates public parameters (a common reference string) for a SNARK. The randomness used must be destroyed ("toxic waste"); if retained, it could be used to forge false proofs.
Ring Signature	A digital signature scheme where a signer signs on behalf of an ad-hoc group ("ring") of possible signers. Verifiers confirm the signature came from a group member but cannot identify which one. Used in Monero.
Stealth Address	A one-time address generated by the sender for each transaction using the receiver's public key. Only the intended receiver can derive the private key to spend funds, breaking on-chain address linkability.
Nullifier	A unique value derived from a spent note or coin that is revealed and recorded on-chain to prevent double-spending, without revealing which note was spent. Core to Zcash's shielded transaction protocol.
ZK-Rollup	A Layer-2 scaling solution that bundles many transactions off-chain and submits a single ZK validity proof to the base layer (e.g., Ethereum). Inherits the security of the base layer without requiring data availability assumptions.
zkEVM	A zero-knowledge Ethereum Virtual Machine: a ZK circuit that proves the correct execution of EVM bytecode. Enables ZK-rollups to run existing Ethereum smart contracts without modification.
Commitment Scheme	A cryptographic primitive that binds a prover to a value ("commits") without revealing it, and later allows the value to be opened and verified. Used throughout ZK proof constructions (e.g., Pedersen commitments).
FRI Protocol	Fast Reed–Solomon IOP of Proximity. An interactive oracle proof used in STARKs to prove that a function is close to a low-degree polynomial, enabling transparent (no trusted setup) proof generation using only hash functions.
Verifiable Computation	A paradigm in which a computationally weak verifier can outsource a computation to an untrusted prover and receive a proof