Sepolia = Public Testnet for Eth

Topic :

PAGE NO. Aman
DATE / /

ZK-fundamentals
- Proof without revealing info

Interactive ZK Proofs
- Must go through multiple rounds of interaction
- Time consuming & not practical
- Eg is Peltriks cave.

Non Interactive ZK (NIZK)
- Only require single round of communication
- Single message Proof & Practical
- Eg: Where's Wally one

Types of NIZK

1. SNARKs (Succint Non-Interactive Argument of knowledge)
   - Succint meaning very small in size & quick to verify proofs
2. STARKs (Scalable Transparent Argument of knowledge)
3. Bulletproofs –
   - Known for not requiring trusted setup & for being efficient for proving statement that ranges.

Zero Knowledge

Never Stop Learning...

Terminology

1. Claim / Statement
   - Statement i.e true.
2. Private & Public Inputs
   - Private: Value only known to prover
   - Public: value known to prover & verifier.

[my age] is over the [minimum age].
- my age → Private Input
- minimum age → Public Input

3. Constraint
   - Mathematical expression that must be true

I know x such that [x2 = 9]
- x2 = 9 → constraint

4. Circuit
   - System of constraints makes up circuit i.e relations.

Eg
My age is over minimum age
Here circuit are
- my age > min age
- my age < max age

Never Stop Learning

Topic : .....

PAGE NO. Aarti
DATE / /

(5) Witness
↳ Set of private values that allow a prover to demonstrate their claim is true.

I know x such that $x^2 = 9$

witness
3
-3
2

(6) Proves & verifies.

(7) Trusted Setup
↳ Ceremony or procedure i.e done once to generate some data that must be used every time some cryptographic (ZK) protocol is run
↳ Toxic Waste = During Trusted setup, random values are generated. They are properly & verifiably destroyed after public parameters are created which is toxic waste.
↳ Common Reference String (CRS)
↳ set of public parameters that both P & V use in proof gen & verification process

Topic : .....

PAGE NO. 4
DATE / /

↳ Structured Reference String (SRS)
  ↳ Type of CRS where public parameters have particular mathematical structure like eg. τ (Tau)

↳ Multi-Party Computation (MPC)
  ↳ To mitigate single party from generating secret, many trusted setup deploy MPC. which is cryptographic technique that allows multiple parties to collaboratively compute function over their inputs while keeping those input private.

↳ Powers of Tau
  ↳ Refers to specific type of trusted setup ceremony (& resulting SRS), commonly used in SNARK like Groth 16 & PLONK. The SRS generated contains a series of elliptic curve points representing successive powers of secret tau, such as $G_1, \tau \cdot G_1, \tau^2 \cdot G_1, \ldots \tau^n \cdot G_1$ where G_1 is generator point of elliptic curves.

Never Stop

Topic : .....

PAGE NO. Aarti
DATE / /

↳ Polynomial commitment Scheme =
↳ Cryptographic tool that allows a proover to commit to polynomial P(x) in way it hides its coefficient, yet allows them to later prove certain properties abt P(x) (like its evaluation at specific point) without revealing one entire polynomial.
↳ Type of (PCS) & is KZG commitment

Trusted setup eg
↳ can be either (circuit specific) OR
↓
needs to be generated for every circuit.
Universal
↓
Cryptographic parameters can be reused.

Groth16 setup
↳ widely used zk-SNARK that requires circuit-specific trusted setup.
Phase 1: Powers of Tau (Universal component):
↳ The phase generates a general purpose SRS consisting of encrypted powers of (G1, G2, ...)
↳ this part of setup is not specific.

Never Stop Learning.

PAGE NO.
DATE / /

Topic : .....

Phase 2: Circuit-Specific Transformation

- SRS generated in Phase I is then taken & combined with mathematical
- This phase transforms generic SRS into circuit specific proving key (PK) & verification key (VK).
- Importantly, this phase introduces toxic waste.
- Phase 2 is repeated for new circuit.

PLONK Setup — Universal setup

(Permutations over Lagrange bases for Oecumenical Non interactive Arguments of knowledge).

Proof of knowledge / ZKPOK

- Directly proving possession of certain secret info & present to verifier.

Proof of Computation / ZKPOC

- Proves demonstrates that they have correctly executed a specific computation according to a pre-defined set of rules.
- ZKPOC is based on the computation rather than informat

ZkKYC — ZkSync

Topic :

PAGE NO. DATE / /

Requirement for ZkP.

1. Completeness
2. Soundness
3. Ofc should be Zk.

Zk in Practice

- Front End
  - Constraint System where problem is defined mathematically (circuit)
- Back-End
  - Proving System (Noir or circom)

Front End

- Arithmetization & constraint system
  claim: "I am over 18 → require age ≥ min. age & can enter the club".
  - This process is Arithmetization.
  - Written in Domain Specific Langs (DSL) like noir, circom, Leo.
  - Compile to ACIR or RICS etc
- Generate witness

Backend

- Proof generator
  - Take ACIR → create proof of execution
- Verification

Never Stop Learning