celo

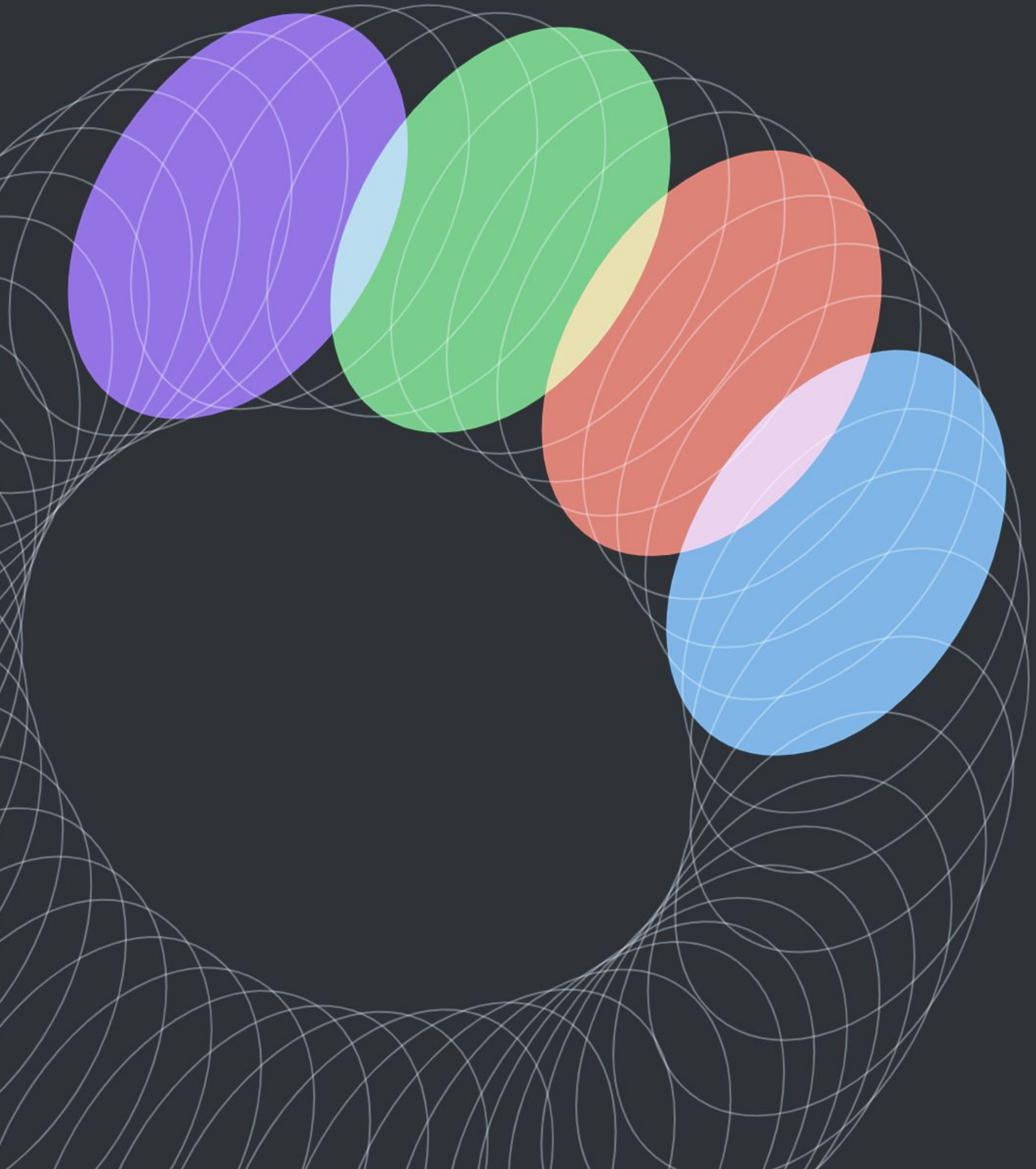
Plumo

Using UltraLight Validation Frameworks to create Fast Syncing Blockchains

Michael Straka
@mstrakastrak

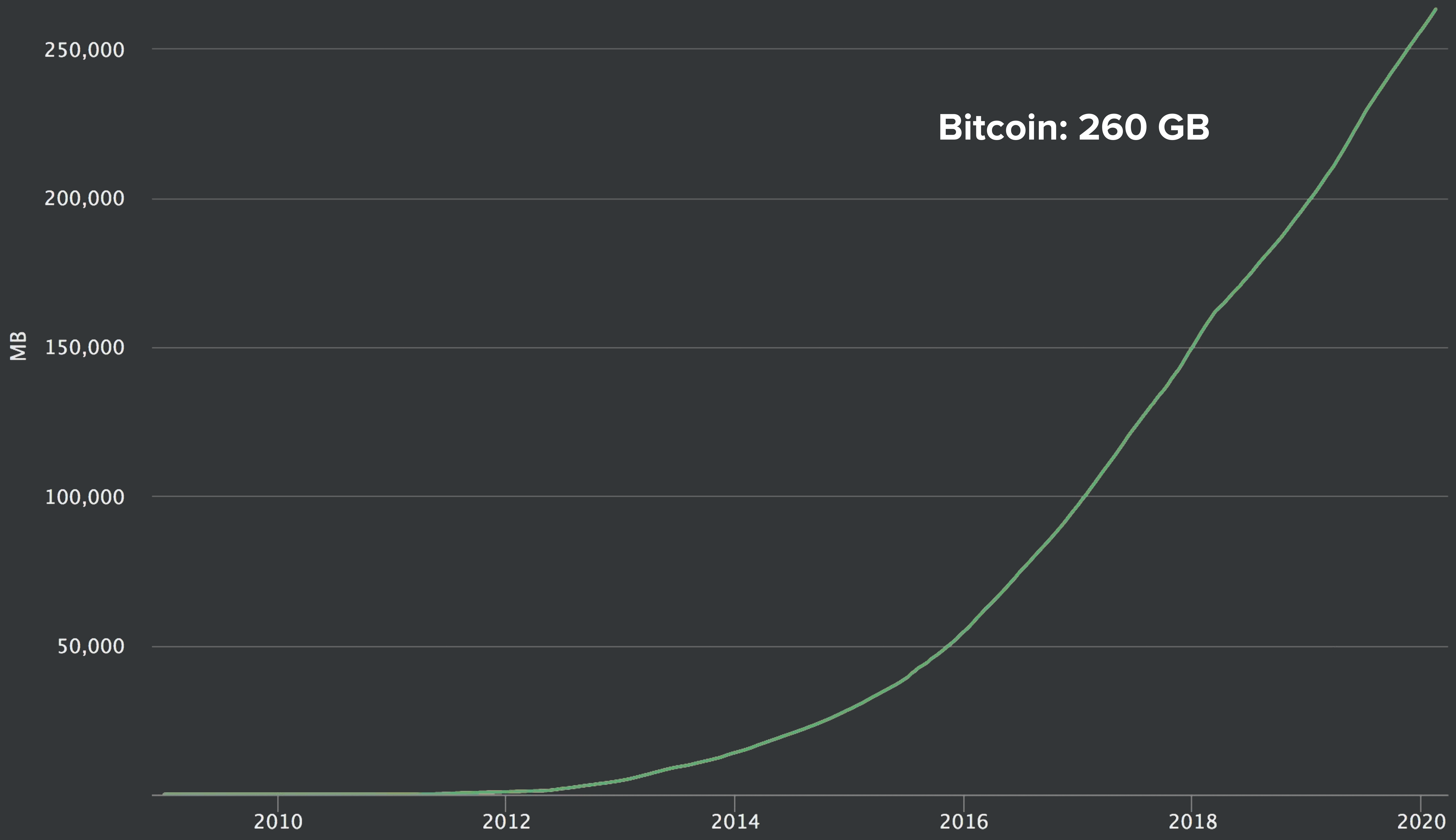
Psi Vesely
@PsiVesely

zkProofs Workshop



What we'll be covering:

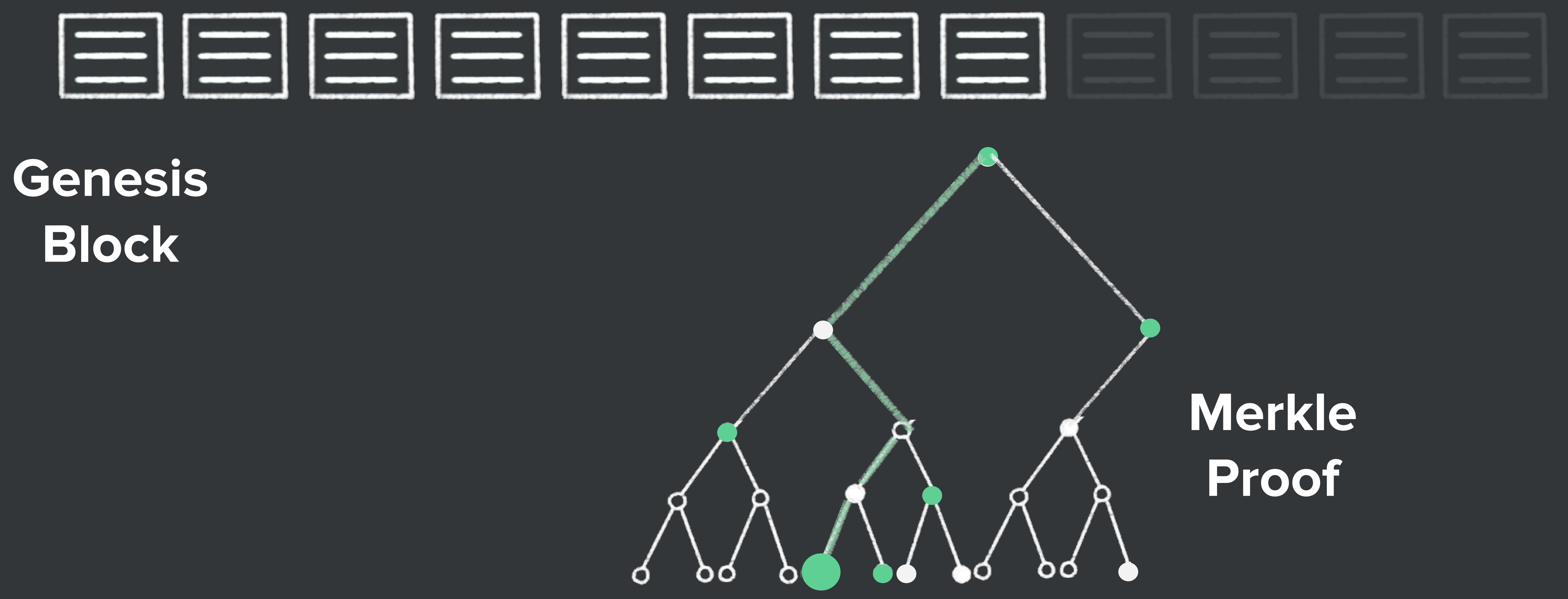
1. Outline of our light client and technical details
2. Standardization candidates:
    a. Cryptographic building blocks
    b. Formal model for ultralight clients



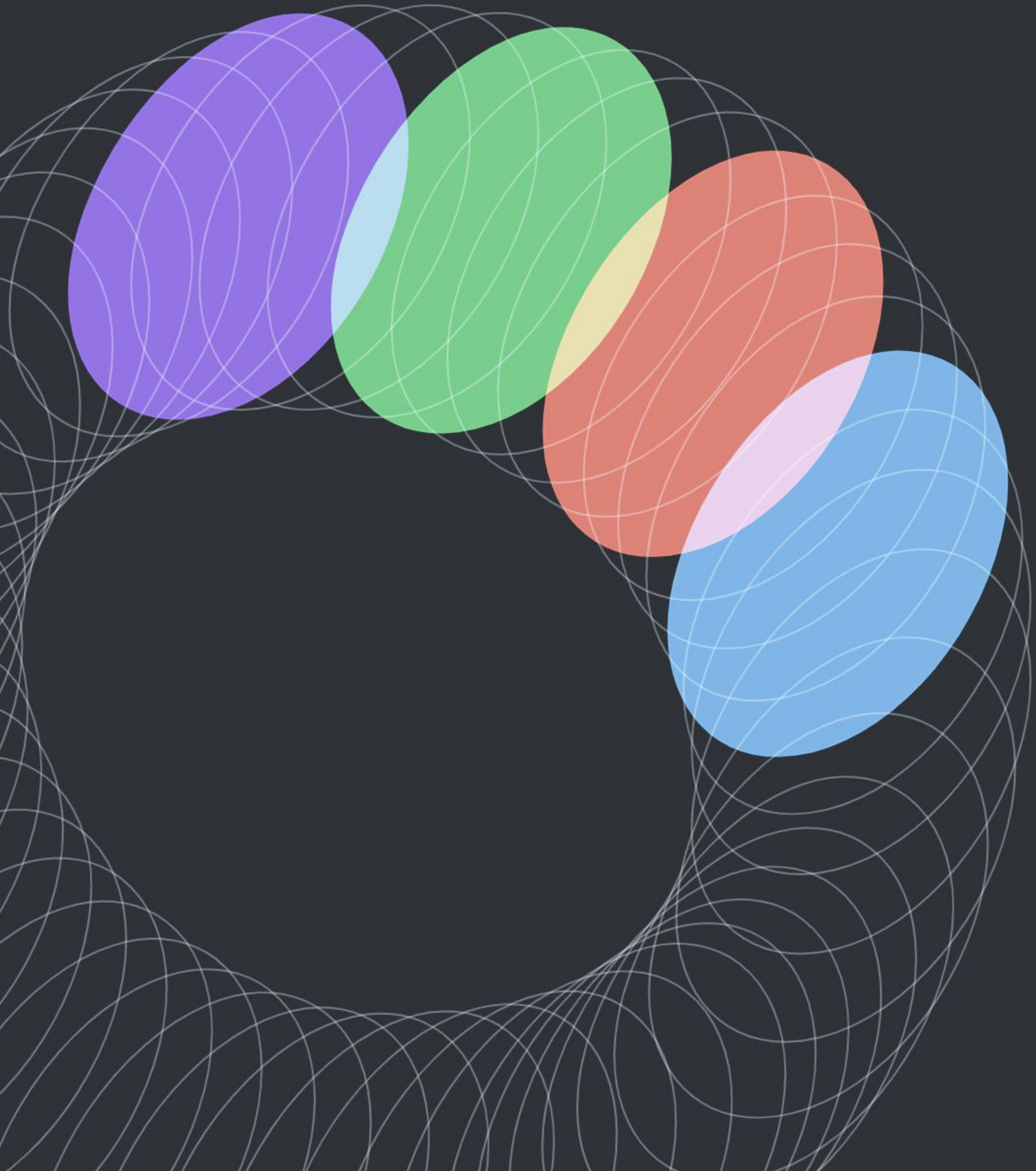
Bitcoin Chain Size

Source: blockchain.com

Simple Payment Verification (Nakamoto 2008)

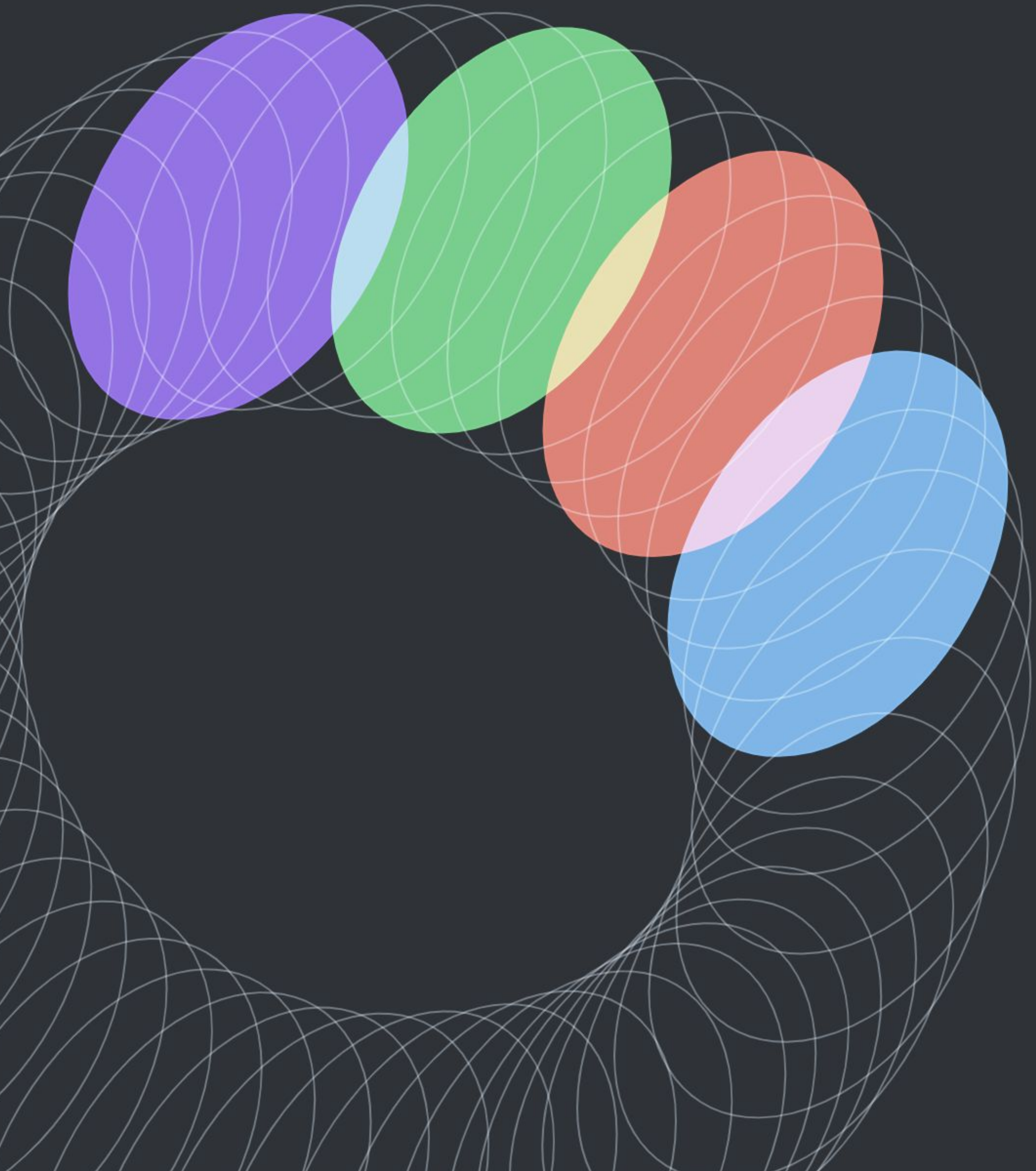


Bitcoin: 47 MB
Ethereum: 4.4 GB



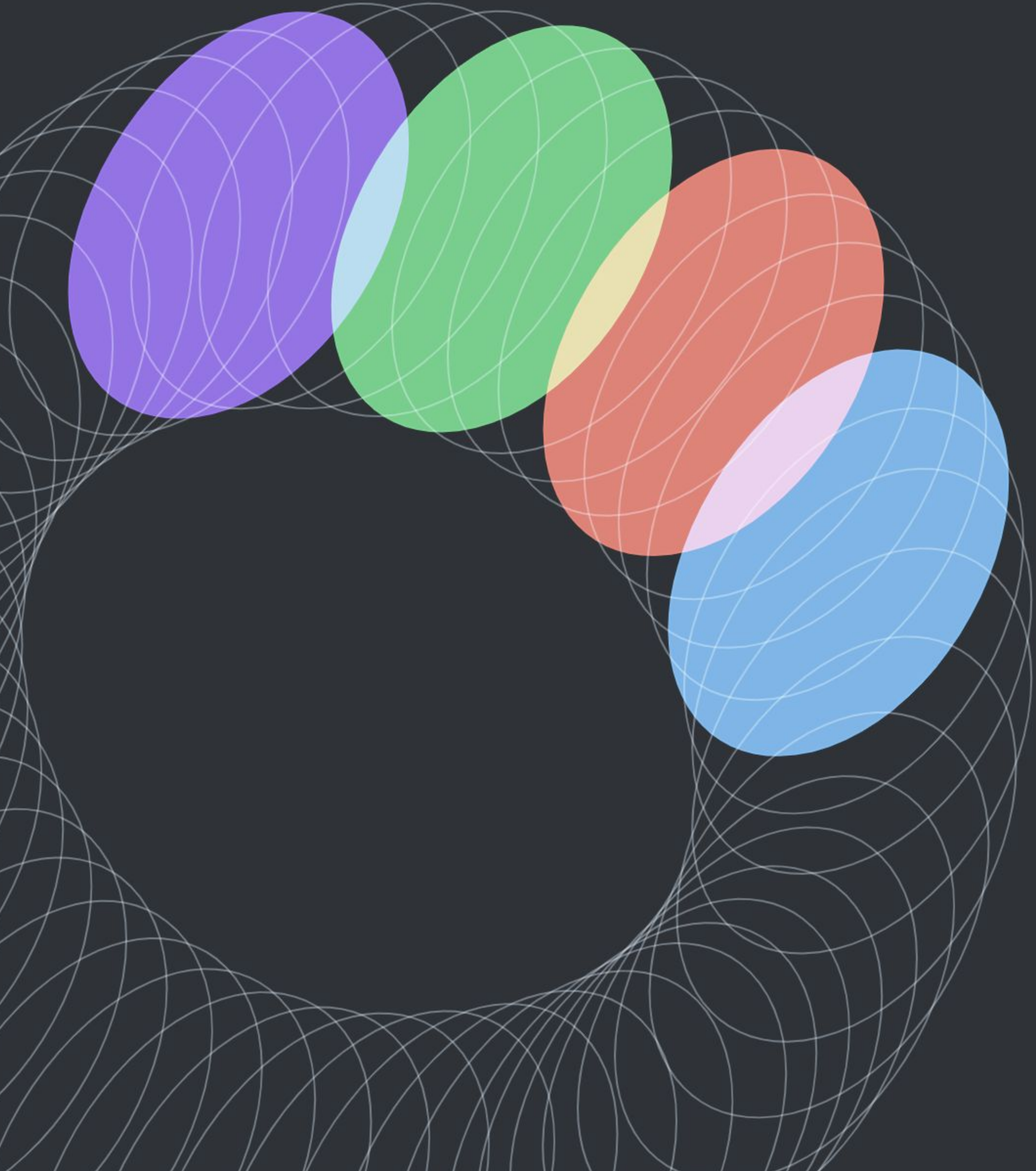
Related Work

1. NiPoPoW (Kiayias, Lamprou and Stouka)
2. Flyclient (Bunz, Kiffer, Luu and Zamani)
    a. These solutions only work for Proof of Work



Related Work

1. Coda (Meckler and Shapiro)
   a. Entire blockchain protocol verified in SNARKs
   b. Proof recursion
   c. Fastest solution for end-user



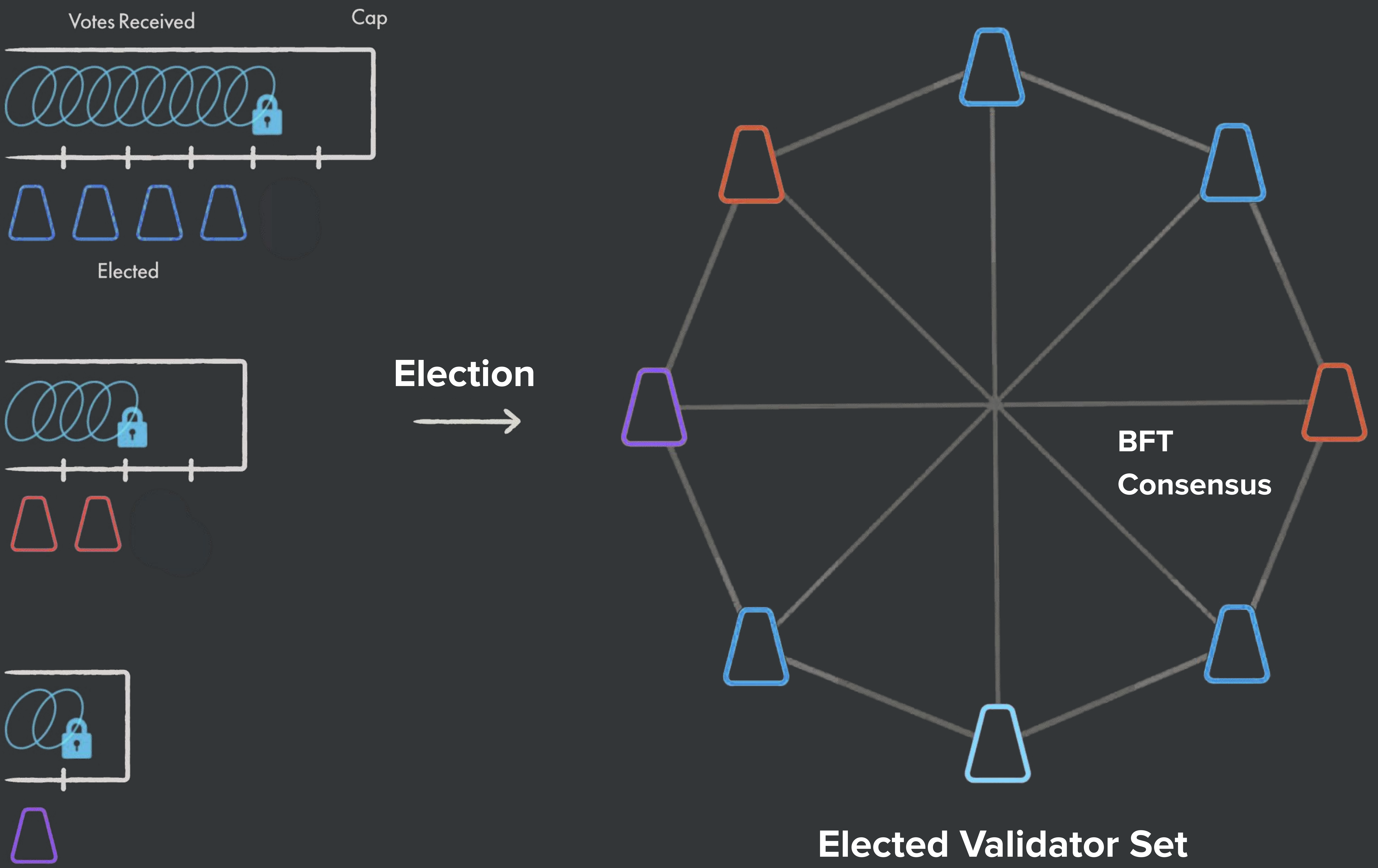
Related Work

1. Coda (Meckler and Shapiro)
   a. Need to modify consensus to be SNARK-friendly
   b. Can get similar efficiency using SPV assumption without proof recursion

Plumo Intro

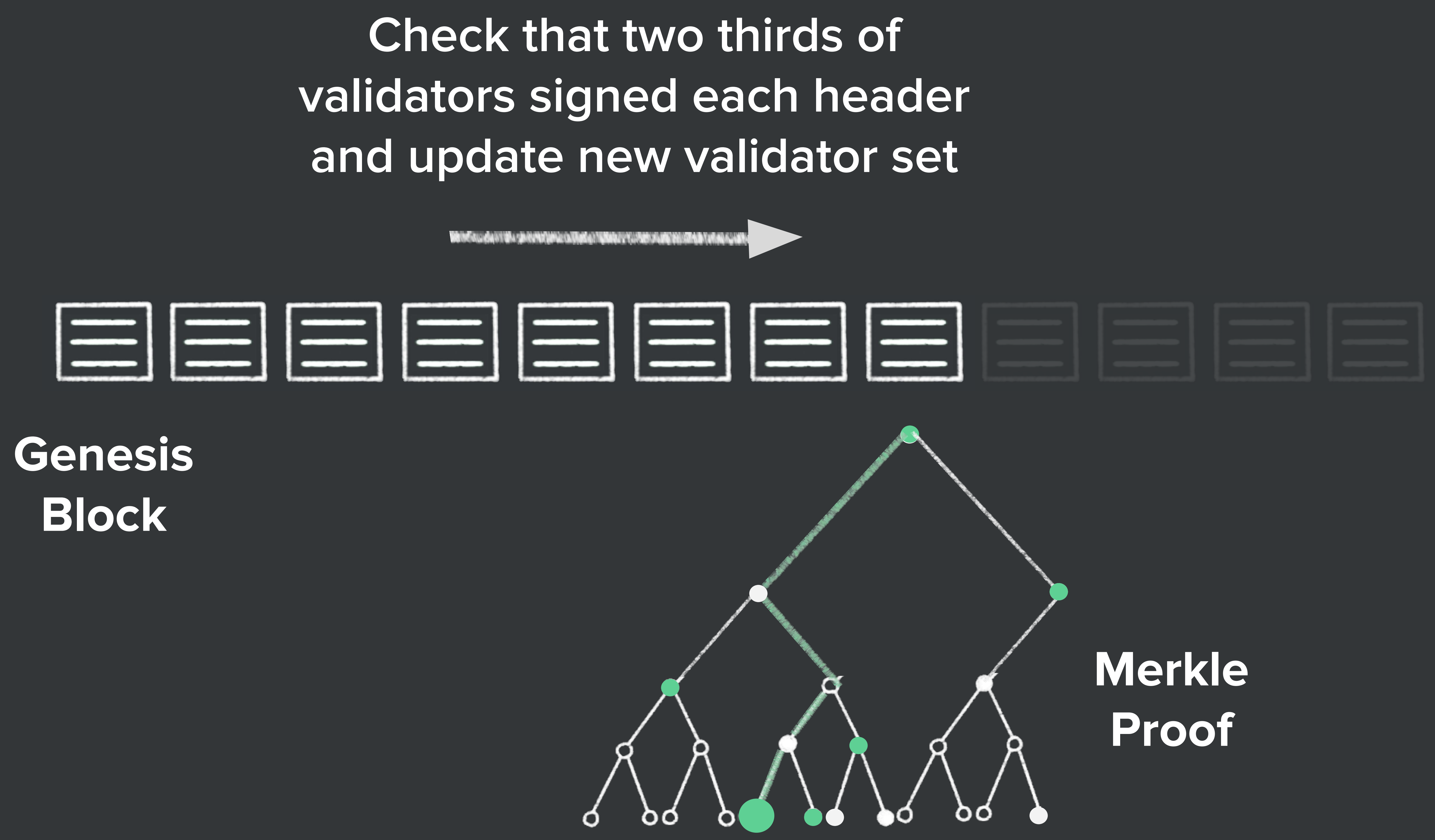
Celo Proof of Stake

Permissionless consensus algorithm
running on decentralized
infrastructure



Elected Validator Set

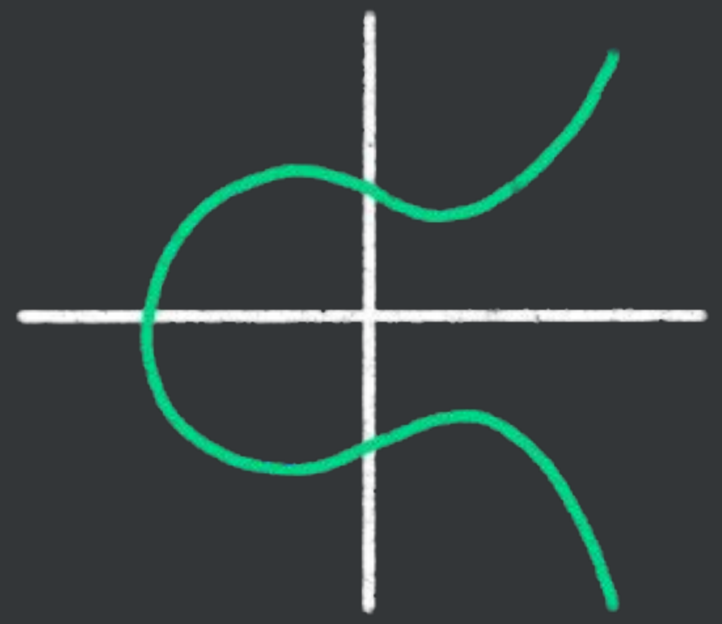
Basic PoS Light Client


Check that two thirds of validators signed each header and update new validator set
Genesis Block
Merkle Proof

Plumo Light Client

Epoch-Based
Syncing

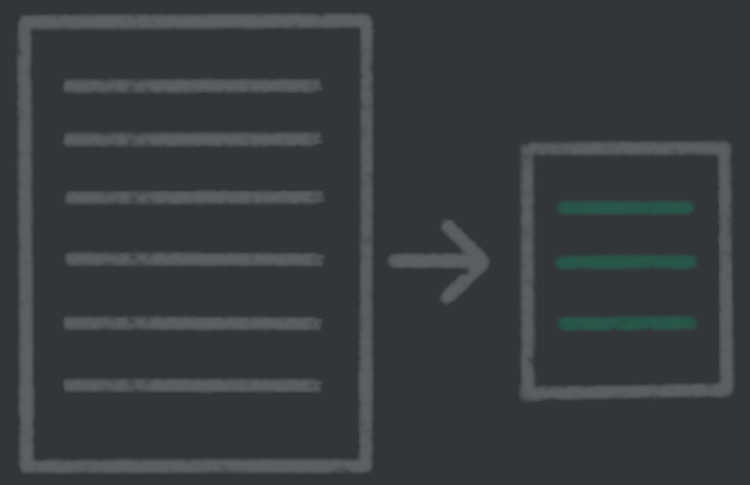
BLS Signature
Aggregation



SNARKs

Plumo Light Client

Epoch-Based Syncing



BLS Signature Aggregation

SNARKs

Plumo Light Client

Epoch-Based Synching

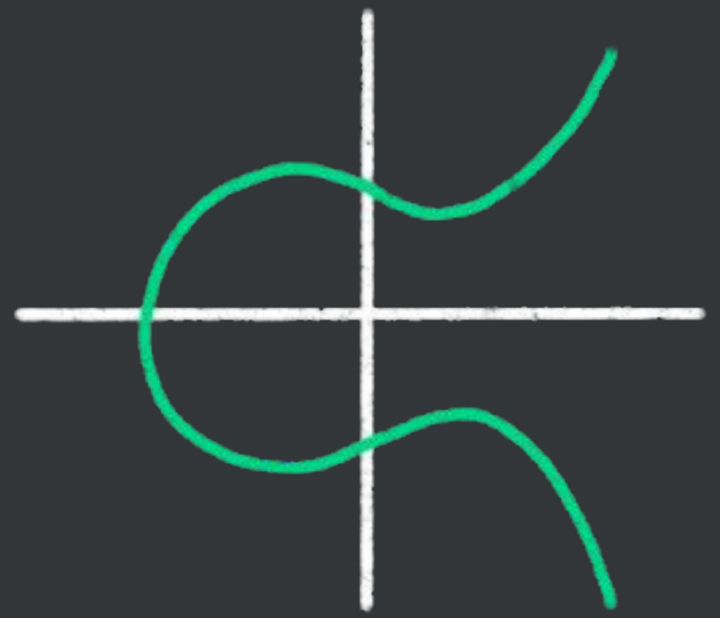
BLS Signature Aggregation

SNARKs

Plumo Light Client

Epoch-Based Synching

BLS Signature Aggregation



SNARKs

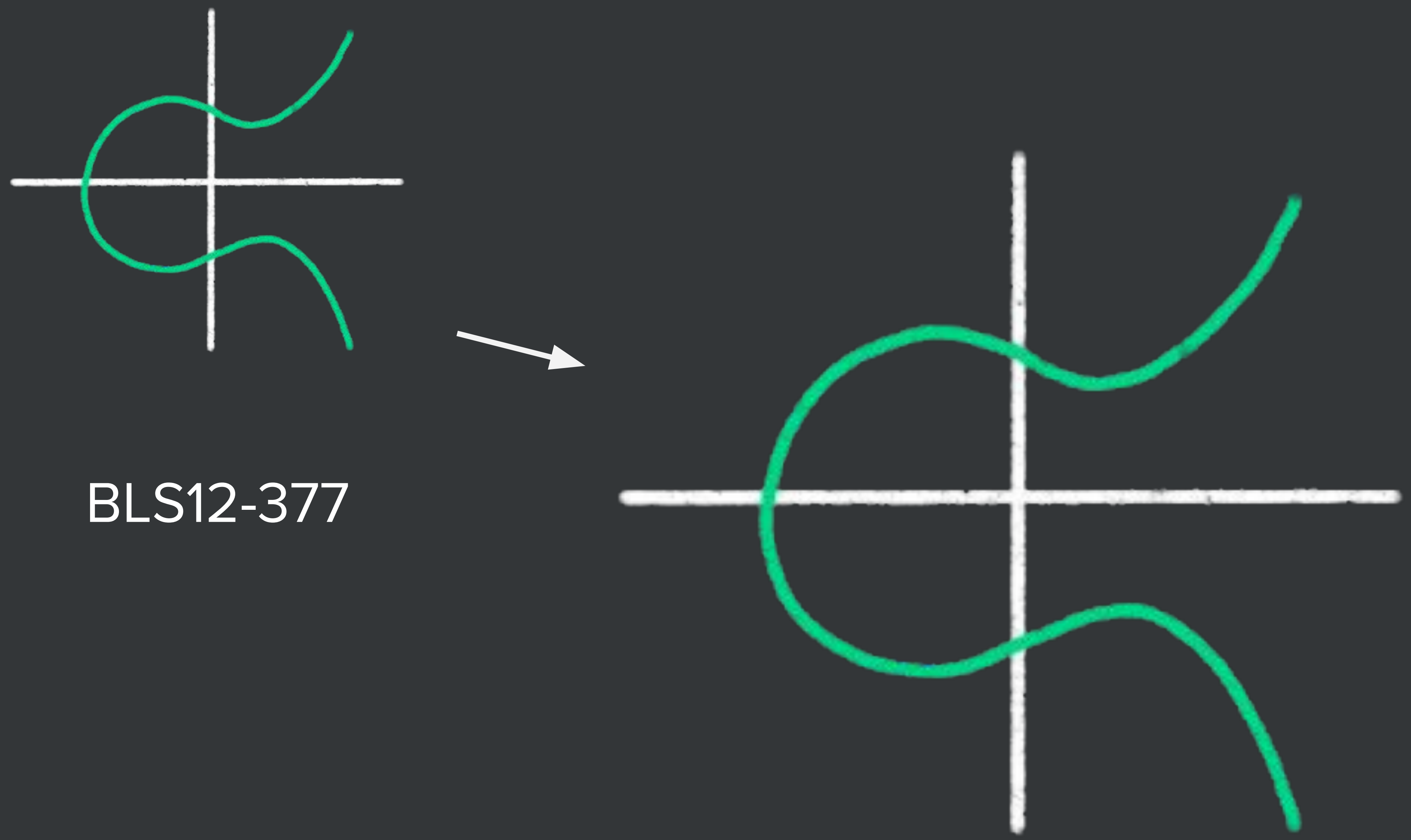
BLS Verification

$$e(\sigma, g) =? \; e(H(m), g^x)$$

2-Chain of Curves

BLS12-377

2-Chain of Curves

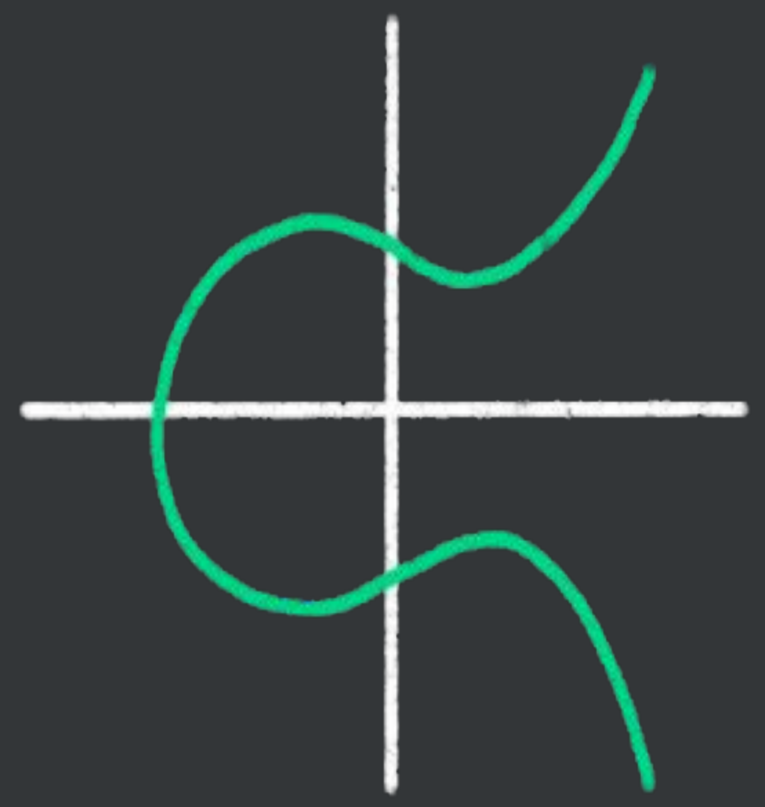


BLS12-377

SW6

BLS Verification

$$e(\sigma, g) =? \; e(H(m), g^x)$$



2-Chain of Curves

	BLS12-377	SW6	BW6
G1 bit size	384	832	768
G2 bit size	768	2496	768

Hybrid Hash Function

$$g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n} \rightarrow y$$

Hybrid Hash Function

$$g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n} \to y$$

$$H(y) \to z$$

Hybrid Hash Function

$$g_1^{x_1} g_2^{x_2} \dots g_n^{x_n} \rightarrow y$$

1. Set i = 0
2. Hash message, nonce pair m | i using Bowe-Hopewood Pedersen hash

$$H(y) \rightarrow z$$

Hybrid Hash Function

1. Set i = 0
2. Hash message, nonce pair m | i using Bowe-Hopewood Pedersen hash

$$g_1^{x_1} g_2^{x_2} \dots g_n^{x_n} \to y$$

3. Feed smaller input into Blake2Xs using XOF to get random-looking 512 bits
4. Interpret bits as x coordinate. Attempt to derive y; if successful return (x,y), otherwise repeat 1-4 with i <- i+1

$$H(y) \to z$$

Other hash-to-curve options?

$$g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n} \rightarrow y$$

1. Try-and-increment not constant time...
2. But BLS12-377 base field prime p = 1 mod 4 so square roots stuck with Tonelli-Shanks, also not constant-time

$$H(y) \rightarrow z$$

Other hash-to-curve options?

$$g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n} \rightarrow y$$

1. Try-and-increment not constant time…
2. But BLS12-377 base field prime p = 1 mod 4 so square roots stuck with Tonelli-Shanks, also not constant-time

$$H(y) \rightarrow z$$

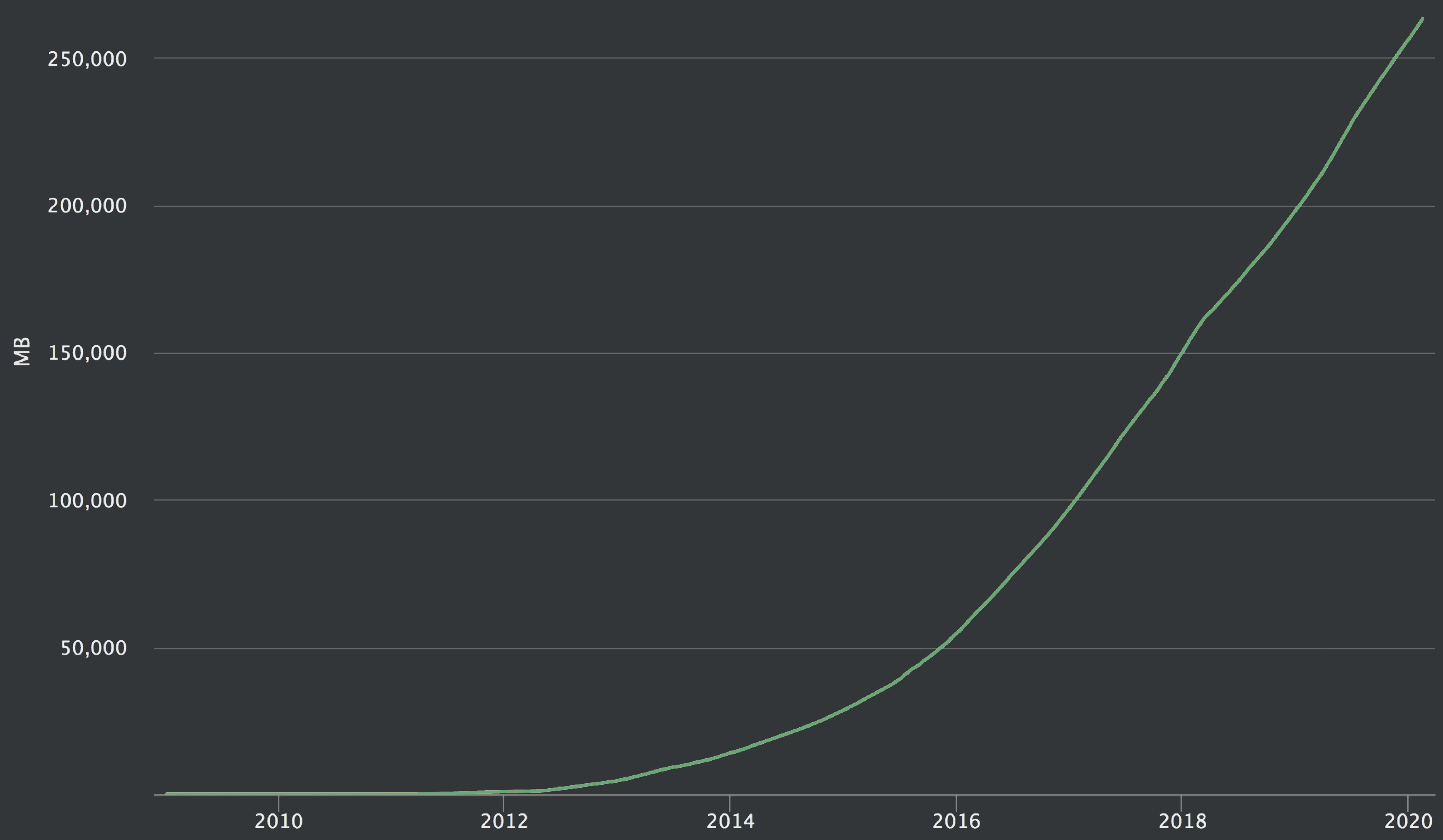
3. SWU mapping optimizations for p = 3 mod 4 possible in BLS12-381 also not possible in 377
4. One extra inversion and Legendre symbol computation each necessary

Constraint Costs

	Miller Loop	Final Exp.	Blake2Xs	Pedersen
Constraints	~4700	~7900	~22000	1.6 / bit

Scaling problem part ii: prover edition

Bitcoin: 260 GB



Bitcoin Chain Size

Source: blockchain.com

Enter Incremental Proofs

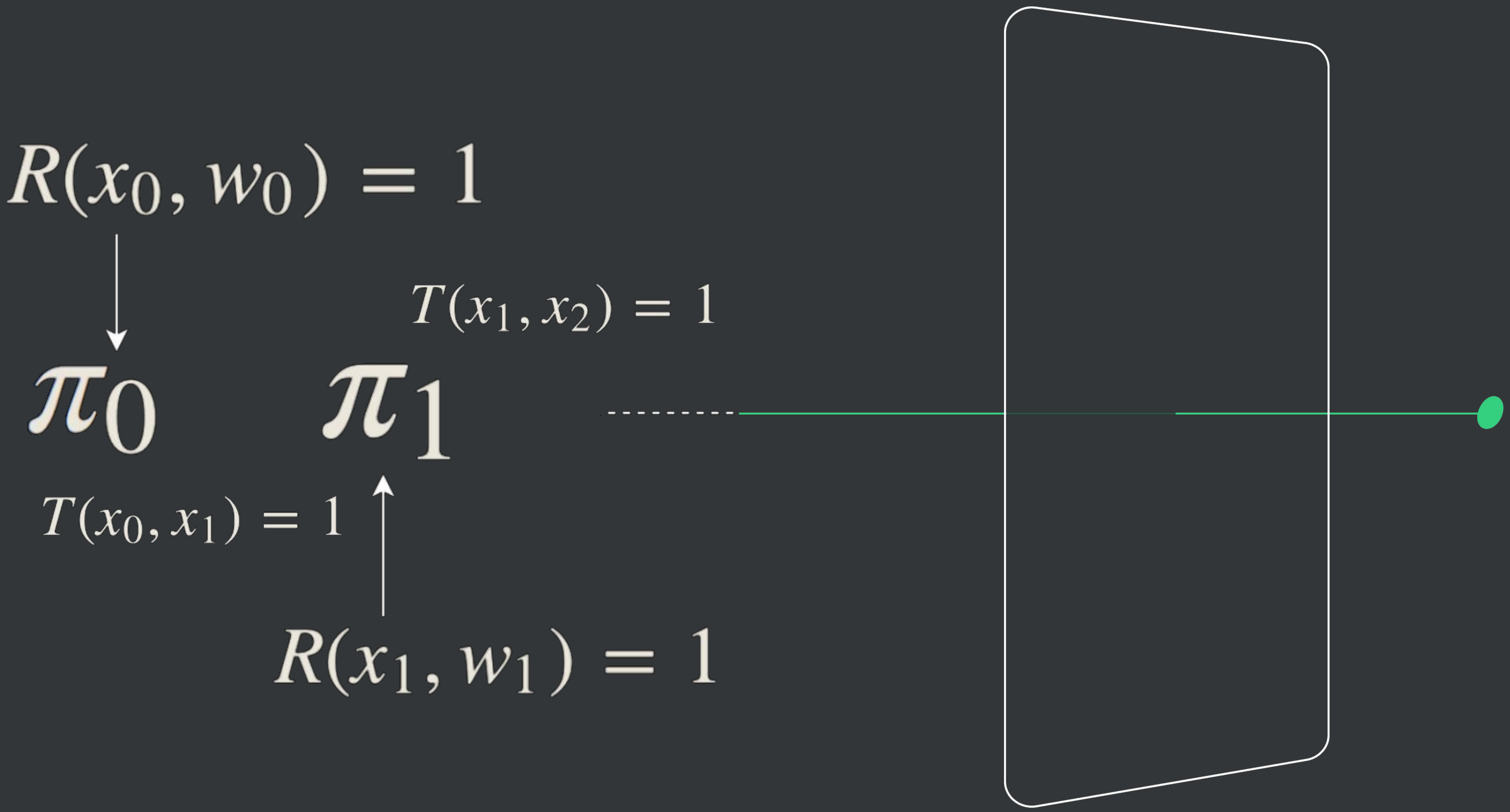
A First Approach: Recursion

$$\begin{tikzcd}
R(x_0, w_0) = 1 \arrow[d] & & & R(x_{n-1}, w_{n-1}) = 1 \arrow[d] & \\
\pi_0 \arrow[r] & \pi_1 \arrow[r] & \cdots \arrow[r] & \pi_{n-1} \arrow[r] & \pi_n \\
& R(x_1, w_1) = 1 \arrow[u] & & & R(x_n, w_n) = 1 \arrow[u]
\end{tikzcd}$$

Simple Inductive Solution

$$\begin{tikzcd}
R(x_0, w_0) = 1 \arrow[d] & & \\
\pi_0 \arrow[r, phantom, "T(x_0, x_1) = 1"'] & \pi_1 \arrow[r, phantom, "T(x_1, x_2) = 1"] & \cdots\cdots \\
& R(x_1, w_1) = 1 \arrow[u] &
\end{tikzcd}$$

Simple Inductive Solution



$R(x_0, w_0) = 1$

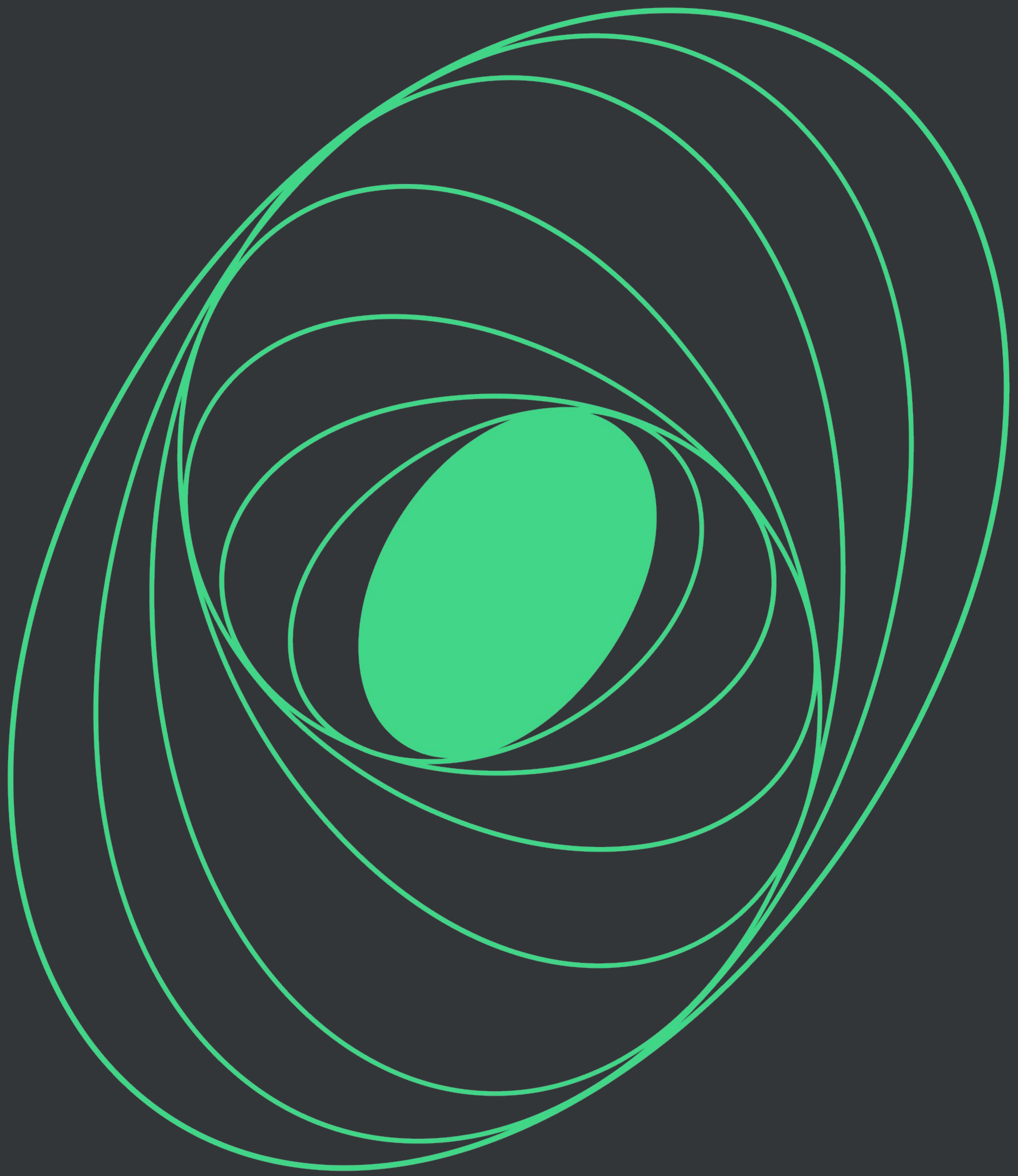
$T(x_1, x_2) = 1$

π_0 π_1

$T(x_0, x_1) = 1$

$R(x_1, w_1) = 1$

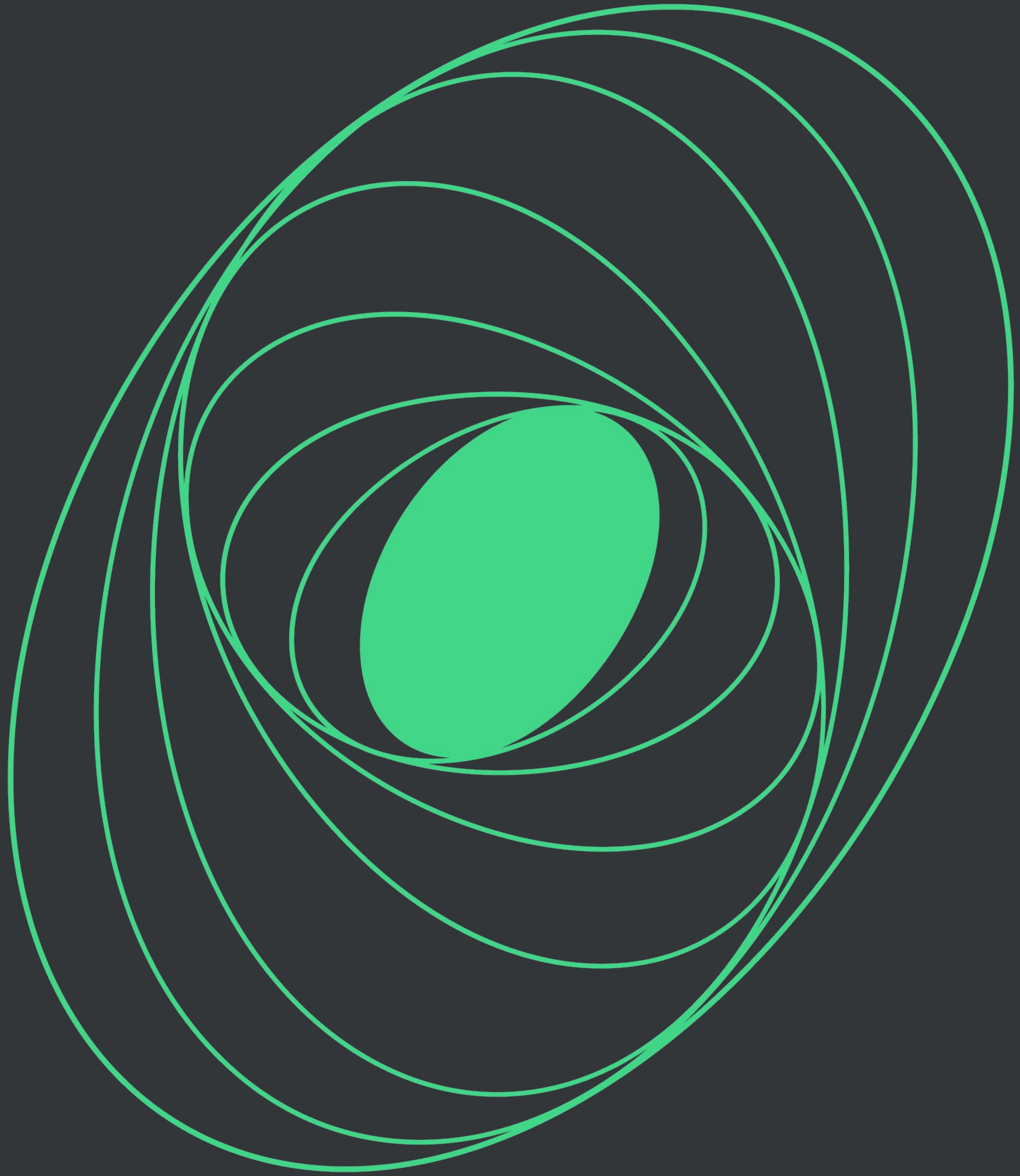
Our SNARK Circuit



What are we proving per epoch?

For each epoch message:

1. The aggregate public key was formed by adding at least ⅔ the current committee's public keys according to the bitmap.
2. The current epoch message number is 1 greater than the last.
3. The multisignature is valid with respect to the message and aggregate public key.



What are we proving for multiple epochs?

For each epoch message:

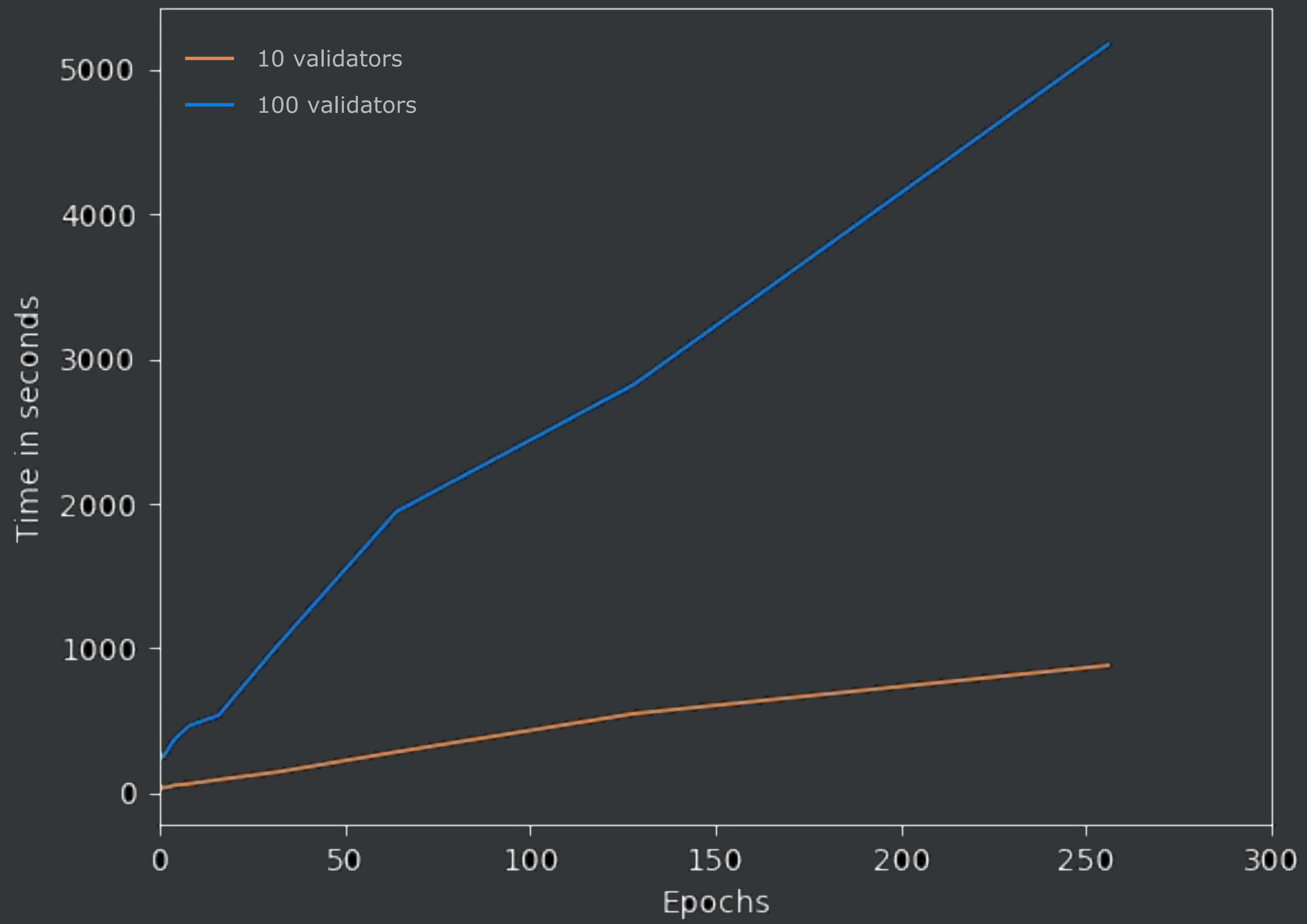
1. The aggregate public key was formed by adding at least ⅔ the current committee's public keys according to the bitmap.
2. The current epoch message number is 1 greater than the last.

Then once:

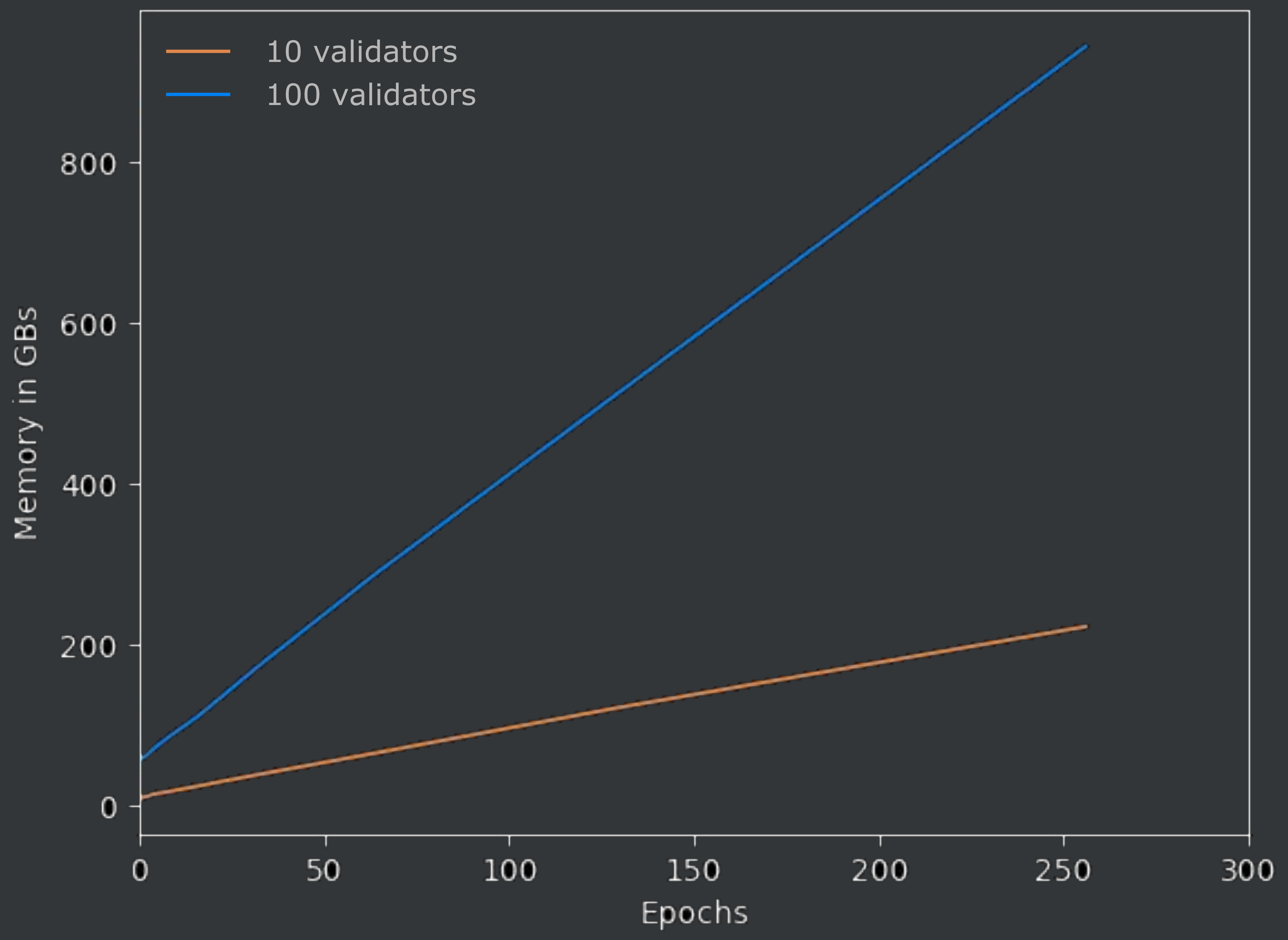
- The aggregate multisignature is valid with respect to the messages and aggregate publics keys.

Performance Results

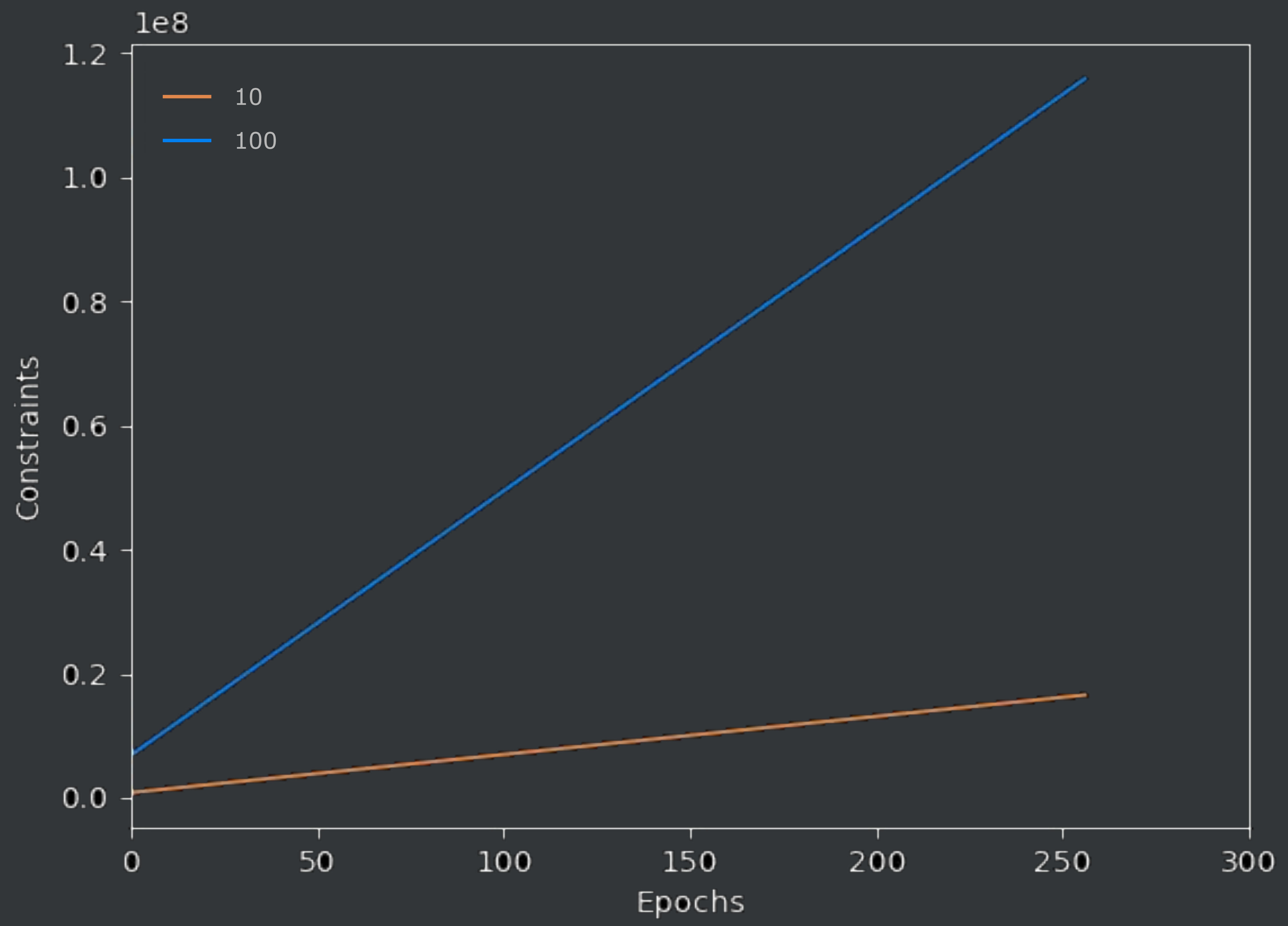
(Proving)



Proof Generation Time



Memory Consumption



Number Constraints

Performance Results

(Verifying)

Verification Times On Mobile Devices

Samsung Galaxy S10	OnePlus 7	Motorola Moto G7	Motorola Moto G2
0.23 s	0.25s	1.53s	6s

Estimated Gas Cost of a Bridge Transaction Between Celo and Ethereum

Validate Plumo Proof[1]	Merkle Proof	Total	Price in USD[2]
4,002,411	< 20,000	< 4,030,000	$6

[1] Assuming EIP 1962

[2] Using gas and ETH prices on Feb 18, 2020

Thanks, we value your time

@mstrakastrak @PsiVesely

Celo | @CeloOrg | celo.org

Discussion: Standardization Candidates

1. SNARK signature verification
   a. 2-chain of elliptic curves. BLS12-377 with BW6 a candidate
2. Ultralight client formalism
3. SNARK-friendly hash-to-curve