ZK Proof Performance and Security Characteristics

Brian Wilkes CFA
Ethereum Foundation PSE Grantee
What's on the Docket?

- Background
- How to find ZKPs on and off chain
- Characteristics & what they mean
What are ZKPs?

Alice (Prover)

I can drink!

Birthday

ZK Circuit

Inputs

Proof

Bob (Verifier)

Show me the Proof!
Why Zero-Knowledge Proofs?

- Scalability
  - 13 TPS to 13K TPS
  - Separate Chains w/ Full Security of Ethereum
- Privacy
  - Ethereum is PUBLIC
Scalability

Layer 2
Blockchain
Block 5

Layer 2
Blockchain
Block 6
Privacy

Secret → ZK Circuit → Proof

Ethereum Main Chain
  Verifier Smart Contract
  Funds Released
The Search for Verifiers

- Important Characteristics of a ZKP are in the Circuit
- Each Circuit has a unique Verifier on-chain
- L1 Chain Data is PUBLIC

How do we find Verifiers on Chain?

zkSync V1 Sept 20–26, 2022

- zkSync Block: 15414
  - 2170 Blocks
- zkSync Block: 17584
  - 265,613 Txs

214 Inputs → ZK Circuit → 214 Proofs → Verifier Smart Contract
BigQuery

- bigquery-public-data
  - crypto_ethereum
    - amended_tokens
    - balances
    - blocks
    - contracts
    - logs
    - sessions
    - token_transfers
    - tokens
    - traces
    - transactions
Bytecode Search for Constants
Finite Field Results

- 558 Contracts with FF Constant
- Checked Every Contract on Etherscan
- Most were not Verfier Contracts
Execution Trace

**Proxy** .83981808() => ()

**ZkSync** .proveBlocks()
(_committedBlocks=, _proof=[{name:recursiveInput, type:uint256[]}, order:1, index:44994363429362], ValueString:11162962296665555999, 2314378739147882166, 14742588698348431398146506, 269414076969748161691, 70196331763518, 1429655530193881289682, 520941787329182719118, 15698816234117272777, 5108525806615991377, 449943631429362], => ()

**Verifier** .verifyAggregatedBlockProof()
(_recursiveInput=[187329679881025715674887764611697255159022732444785648491512, 3935537665114263171926376307, 4935339293445112459511266063262821724285748493829435, 949484721903990214892716307327653470764953317144126, 147462495627716885842, 4488459874513113, 269414076969748161691, 1429655530193881289682, 520941787329182719118, 15698816234117272777, 5108525806615991377, 449943631429362], => (True)

Null Address: 0x000...002 .iffdc7eb() => ()
Null Address: 0x000...005 .000000000() => ()
Null Address: 0x000...005 .000000000() => ()
Method IDs

- Keccak-256(verifyFRI(uint256[],uint256[],uint256,uint256,uint256)
- (0x)e85a6a2847c217651cfe25e4e2df6637ce6c010a99bbf00918...
<table>
<thead>
<tr>
<th>ID</th>
<th>Text Signature</th>
<th>Bytes Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>614653</td>
<td><code>verifyFRI(uint256[], uint256[], uint256, uint256, uint256)</code></td>
<td>0xe85a6a28</td>
</tr>
</tbody>
</table>

https://www.4byte.directory/
65 Confirmed Verifiers on Main Chain
# 16 Active Contracts on Main Chain

<table>
<thead>
<tr>
<th>Project</th>
<th>Latest Deployed Verifier</th>
<th>Proving Sys</th>
<th>Firstrin</th>
<th>Lastrin</th>
<th>Number of Verifiers</th>
<th>Avg gas Per Verification</th>
<th>Libraries</th>
<th>Language</th>
<th>Hashes Used</th>
<th>Purpose</th>
<th>L2?</th>
<th>Rollup or V0?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Finance</td>
<td>0x50a0abd217540925a3e378f4c1f3124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Pedersen</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>zkSync V3</td>
<td>0x7b804038d55788647a97458c40ca2b1c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bellman</td>
<td>franklin</td>
<td>RUST</td>
<td>Rescue, Sh256</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Aztec Connect</td>
<td>0x782b00e53306c5e4b707c1f4076d698c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>barrenberg</td>
<td>C++</td>
<td></td>
<td>Private Defi</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>StarkEx (DYDX)</td>
<td>0xb24036deee4708c3723e0b52d1cd88c9</td>
<td>2022-10-02 15:19:59 UTC</td>
<td>909228</td>
<td>214430</td>
<td></td>
<td></td>
<td>cairo</td>
<td>C++, Cairo</td>
<td></td>
<td>Pedersen</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Messari 1 Ether</td>
<td>0x93e50717283e8aa28d8df543d0f885f0</td>
<td>2022-07-01 01:47:02 UTC</td>
<td>20</td>
<td>219023</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Perpetual and Tr</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Messari 1 Ether</td>
<td>0x3f23984d15b10b03c602c32a2b1f03</td>
<td>2022-05-30 21:23:19 UTC</td>
<td>130</td>
<td>242905</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>No</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>ZK Space</td>
<td>0x44fbed92c265df5e85d56d2</td>
<td>2022-06-20 08:25:15 UTC</td>
<td>3057</td>
<td>430766</td>
<td></td>
<td></td>
<td>bellman</td>
<td>franklin</td>
<td>RUST</td>
<td>Rescue, Bridge to L2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Aztec</td>
<td>0x3a997dd609c3b2c752936</td>
<td>2021-12-12 19:50:49 UTC</td>
<td>2280</td>
<td>632387</td>
<td></td>
<td></td>
<td>barrenberg</td>
<td>C++</td>
<td></td>
<td>Privacy</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Slims (MX, Rinoh, Sonic)</td>
<td>0x327447a2664d425aca4e4e67</td>
<td>2022-06-02 01:19:04 UTC</td>
<td>1038</td>
<td>257410</td>
<td></td>
<td></td>
<td>cairo</td>
<td>C++, Cairo</td>
<td></td>
<td>App Specific Val</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Loopring</td>
<td>0x61db99a22e490761e80b66108</td>
<td>2020-11-24 01:57:51 UTC</td>
<td>32035</td>
<td>420399</td>
<td></td>
<td></td>
<td>ethersign</td>
<td>libsnark</td>
<td>C++</td>
<td>Sliding window</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ZK Swap V2</td>
<td>0x48e401945d0609f55906</td>
<td>2021-07-12 13:24:43 UTC</td>
<td>7283</td>
<td>512265</td>
<td></td>
<td></td>
<td>bellman</td>
<td>franklin</td>
<td>RUST</td>
<td>Rescue, Sh256</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Polygon Hermez</td>
<td>0x10c405100006b0488028</td>
<td>2021-05-13 08:20:21 UTC</td>
<td>4932</td>
<td>202593</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Token Transfer</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Polygon Hermez Withdraws</td>
<td>0x49441e995c854c4e541d9</td>
<td>2021-09-10 08:15:56 UTC</td>
<td>266</td>
<td>202593</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Poseidon</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WanChain</td>
<td>0x842715481e0e8979626c1d</td>
<td>2020-11-02 06:46:24 UTC</td>
<td>5629</td>
<td>15881</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Poseidon</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tornado.Cash</td>
<td>0x1ce172e1200c0b3728c896</td>
<td>2020-05-13 06:38:49 UTC</td>
<td>302984</td>
<td>242203</td>
<td></td>
<td></td>
<td>circom</td>
<td>snarkjs</td>
<td>circom</td>
<td>Pedersen</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Characteristics
Proving Systems

GROTH16 (2016)
TC, Hermez, Loopring

PLONK (2019)
Aztec Connect, zkSNARKs

Halo2 (2020)
Scroll

STARKs

STARK (2018)
Starkware Projects
Trusted Setups (CRS)

- Requires 1 Honest Participant (MPC)
- Where on the Elliptic Curve
Trusted Setup

SNARKs

GROTH16 (2016)
TC, Hermez, Loopring

PLONK (2019)
Aztec Connect, zkSYnc

TurboPLONK (2019)
Aztec

Halo2 (2020)
Scroll

STARKs

STARK (2018)
Starkware Projects
<table>
<thead>
<tr>
<th>Project</th>
<th>Proving System</th>
<th>Trusted Setup?</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recursion

- Parallelization of Proof Generation
- Proof Chaining
Data Availability

zk-Validiums
- State Reconstruction

zk-Rollups

Ethereum L1
- Bridge Contract
- State Root

Bridge Contract
- State Root
- State Reconstruction
Data Availability

- Validiums
  - StarkEx (IMX, Rinofi, Sorare, Apex...)
  - zkPorter
- Rollups
  - zkSync V2
  - Scroll
  - Polygons: Miden, zkEvm, Zero
  - zkSpace
- StarkNet
To Inherit Full Ethereum Security A Rollup Must Have:

- Available Data on L1
- Functional, Accessible Force Exits
- Time Delays for L1 Updates
- L1 Locked Funds = Market Value on L2
Get into the Data

- Big Query (& Dune)
- Project Github
- L2 BEAT
- Layer 2s
Thank You!

Justin Martin (@thefrozenfire)

Mark Roddy (@mroddy5280)
Thank you!

Brian Wilkes
Your title, your organization
e-mail@emailaddress.com
@outsideranalytics

Project Github