

MUTATOR SETS

AND THEIR APPLICATION TO SCALABLE PRIVACY

Alan Szepieniec

艾伦·余丕涅茨

alan@neptune.cash

Neptune

Thorkil Værge

thor@neptune.cash


Neptune



neptune



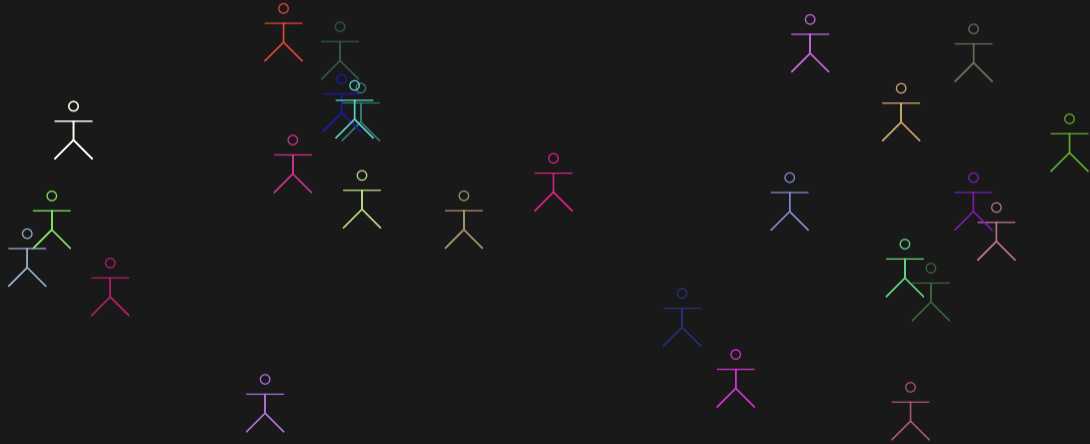
Triton VM

Find Waldo 

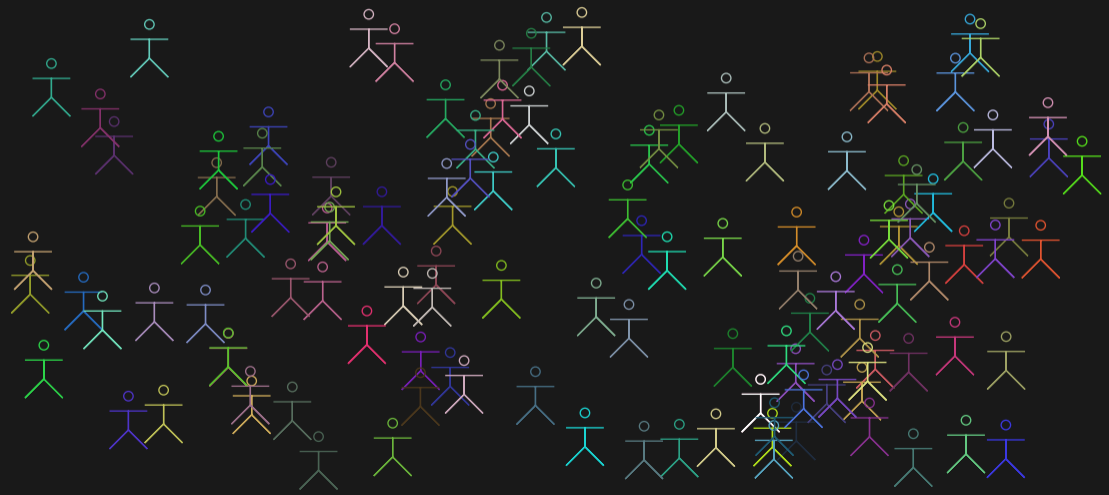
Find Waldo



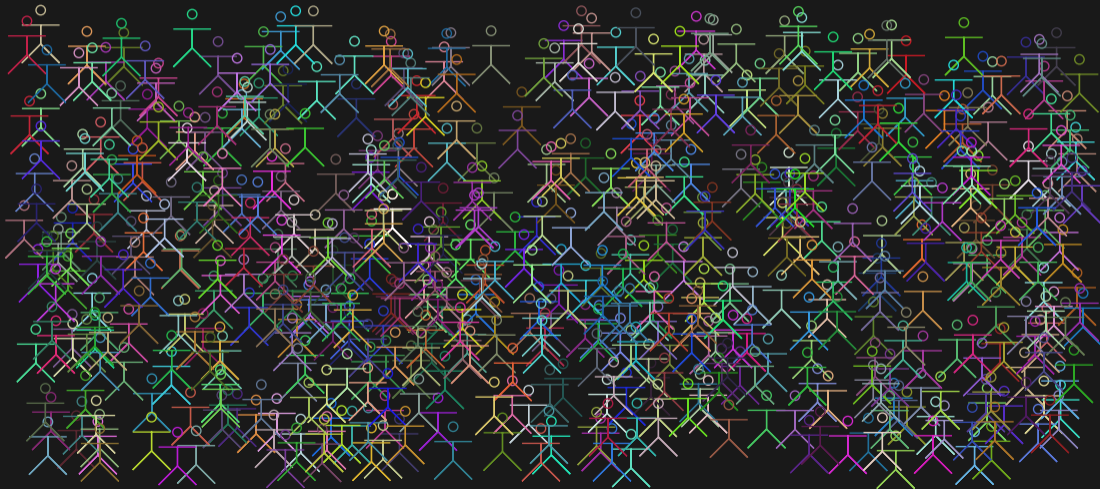
Find Waldo



Find Waldo



Find Waldo



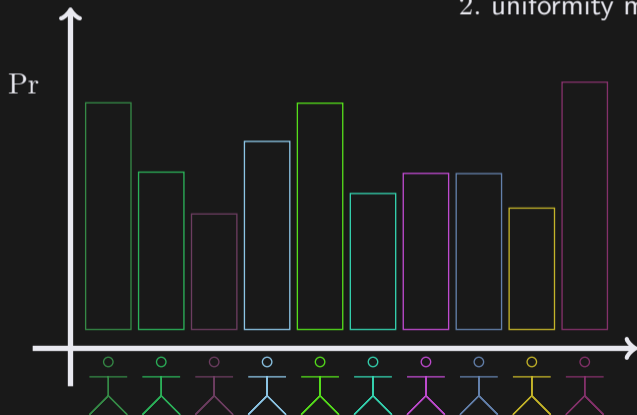
Find Waldo

1. crowds matter
2. uniformity matters

Find Waldo

1. crowds matter

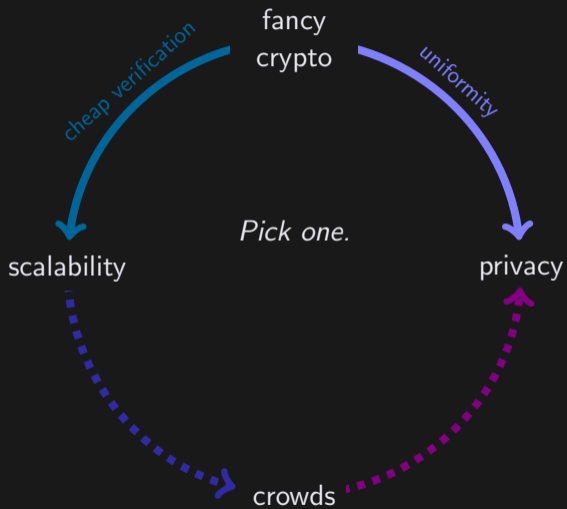
2. uniformity matters



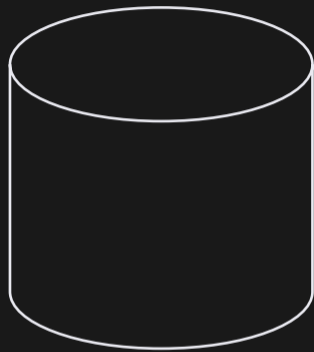
Dilemma



Dilemma

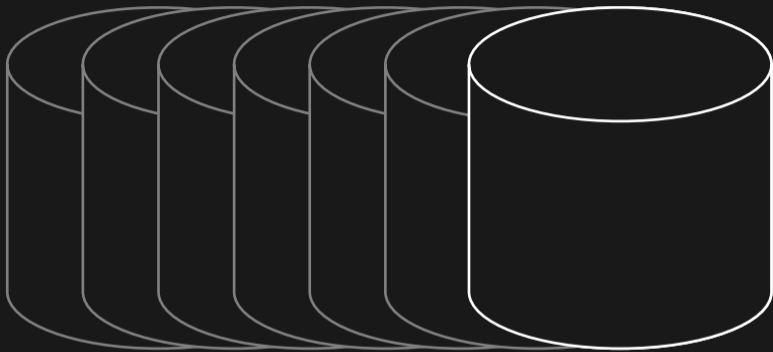


UTXO Set



database of all spendable coins

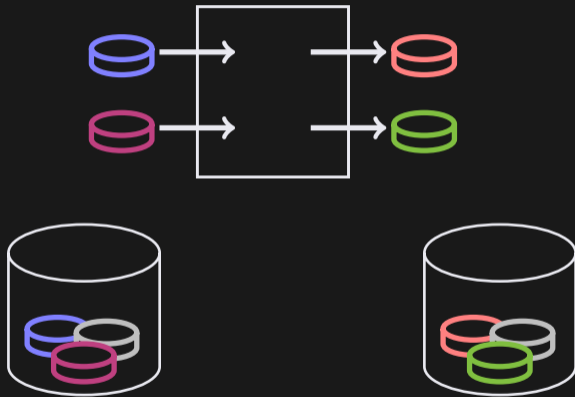
UTXO Set



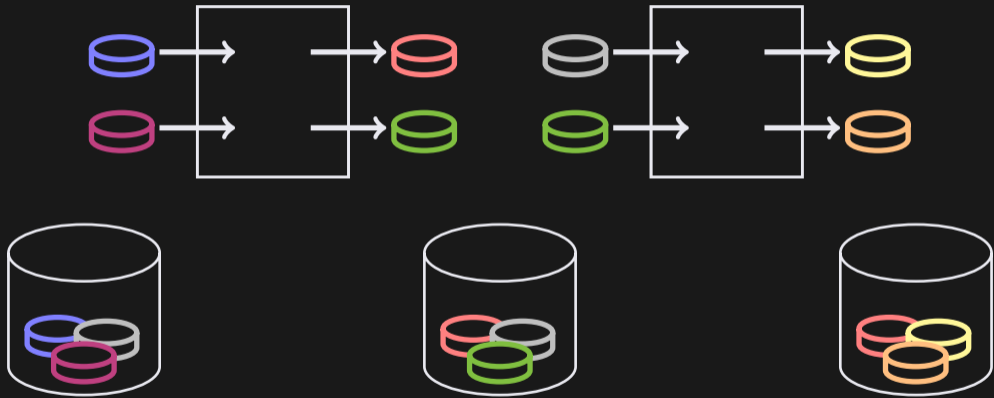
current

database of all spendable coins

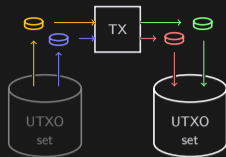
Blockchain Transactions



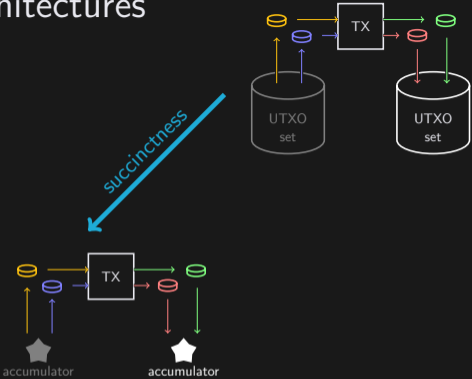
Blockchain Transactions



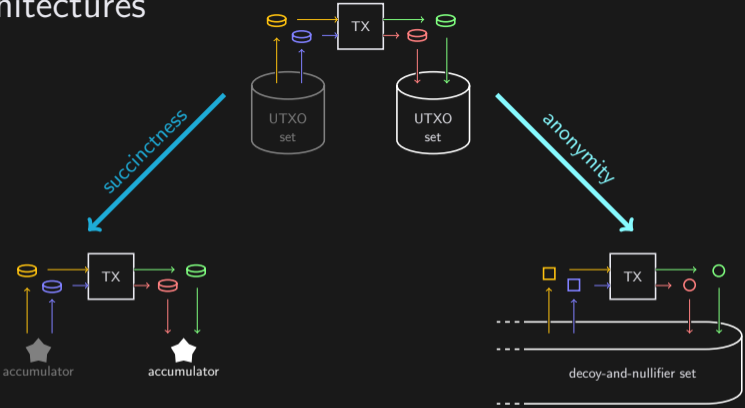
Ledger Architectures



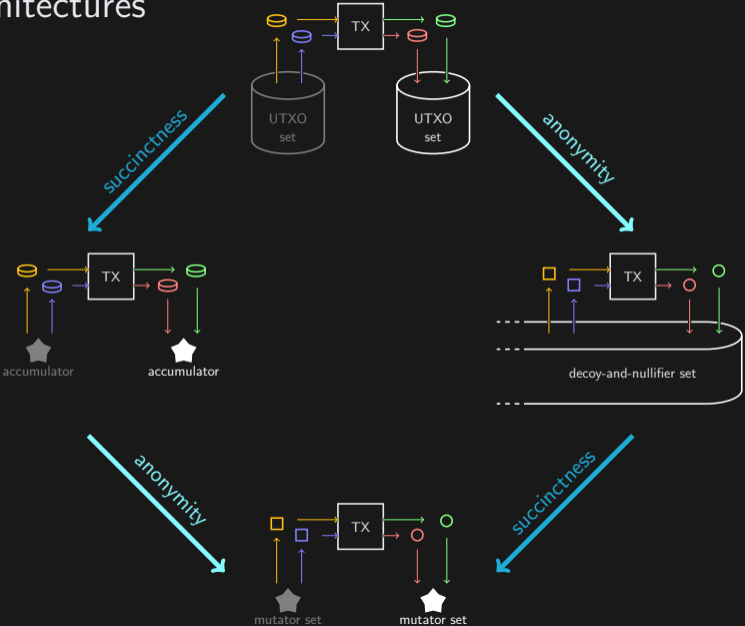
Ledger Architectures



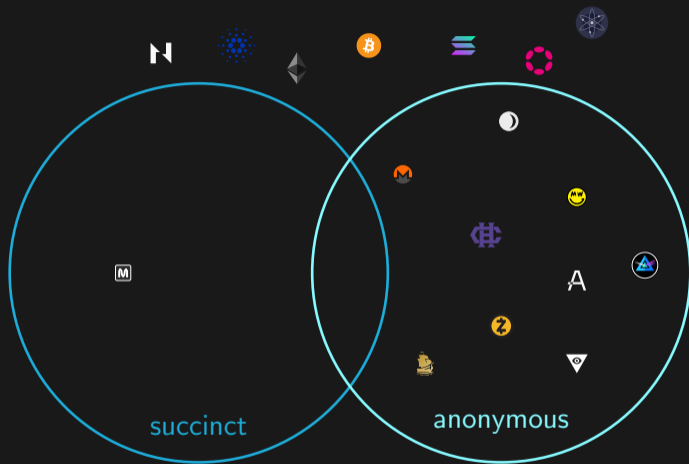
Ledger Architectures



Ledger Architectures



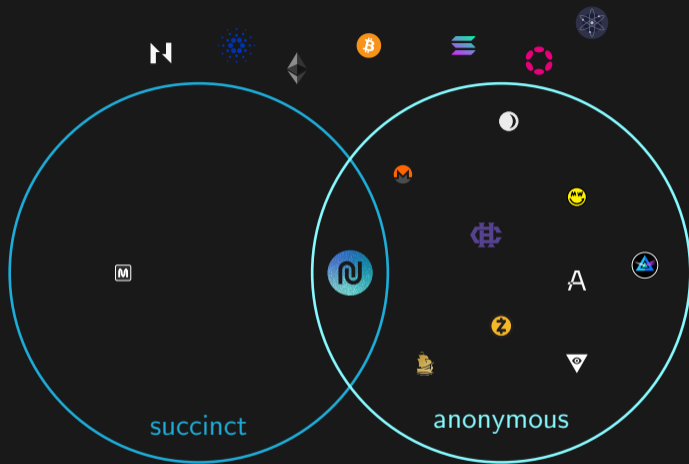
Succinctness \cap Anonymity



negligible cost to
run a full node

transactions are
confidential

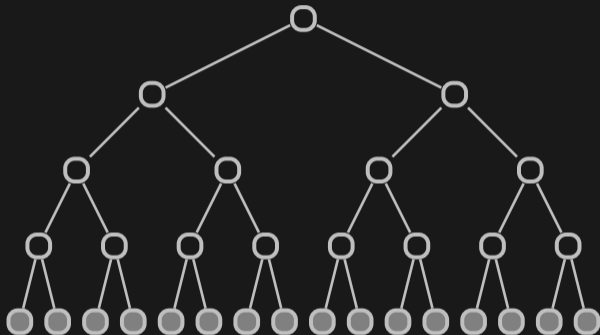
Succinctness \cap Anonymity



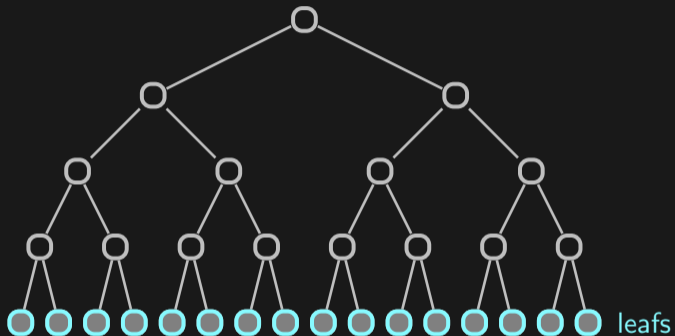
negligible cost to
run a full node

transactions are
confidential

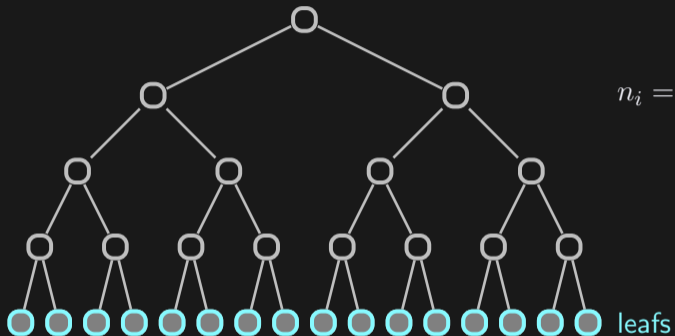
Merkle Tree



Merkle Tree

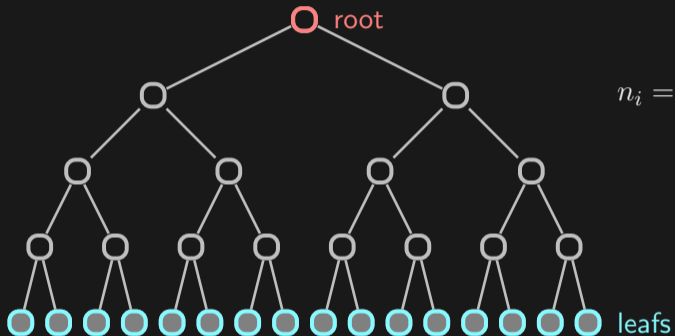


Merkle Tree



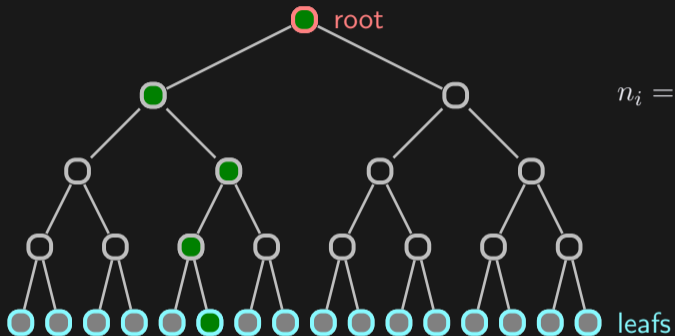
$$n_i = \mathbf{H}(n_{i||0} \parallel n_{i||1})$$

Merkle Tree



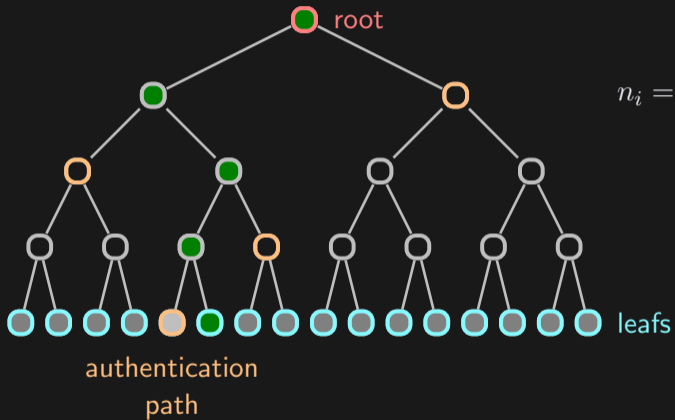
$$n_i = \text{H}(n_{i||0} \parallel n_{i||1})$$

Merkle Tree



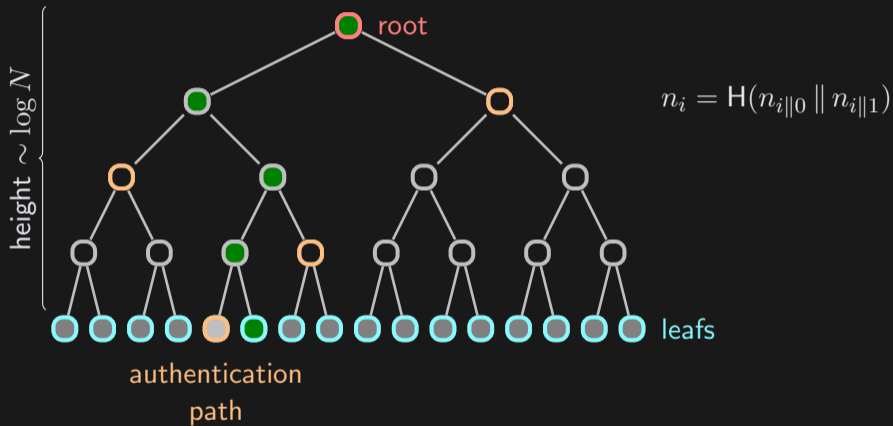
$$n_i = \text{H}(n_{i||0} \parallel n_{i||1})$$

Merkle Tree



$$n_i = H(n_{i||0} || n_{i||1})$$

Merkle Tree



Merkle Mountain Range

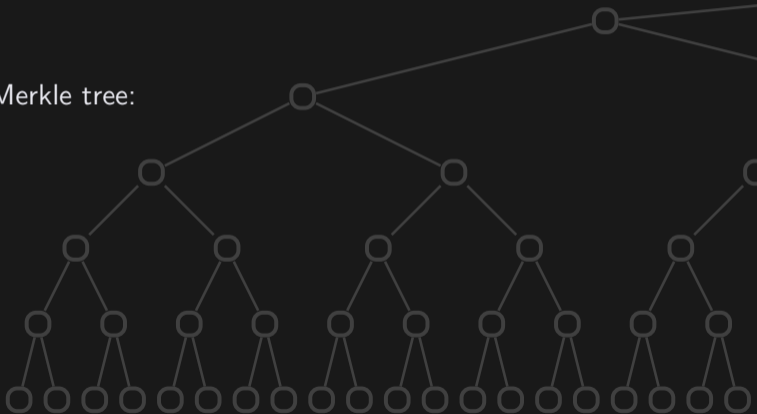
= sequence of progressively smaller Merkle trees

Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible

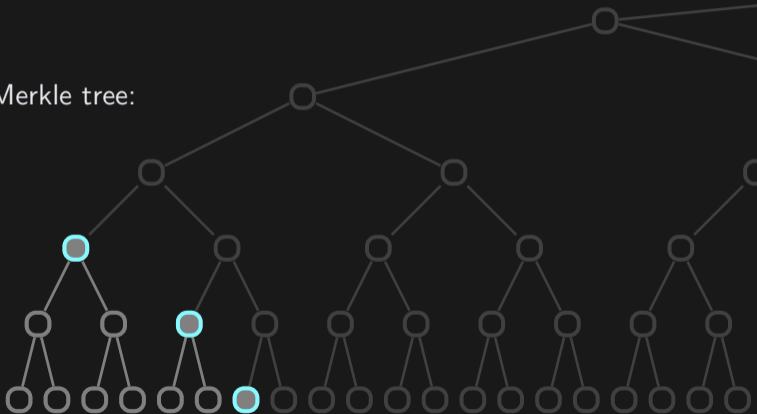


Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible

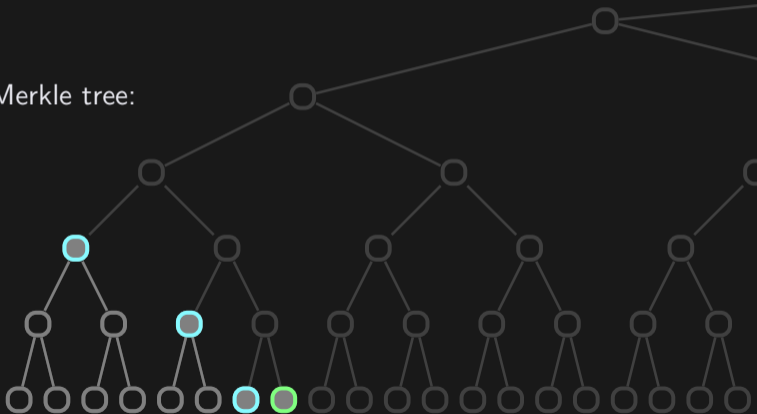


Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget whenever possible

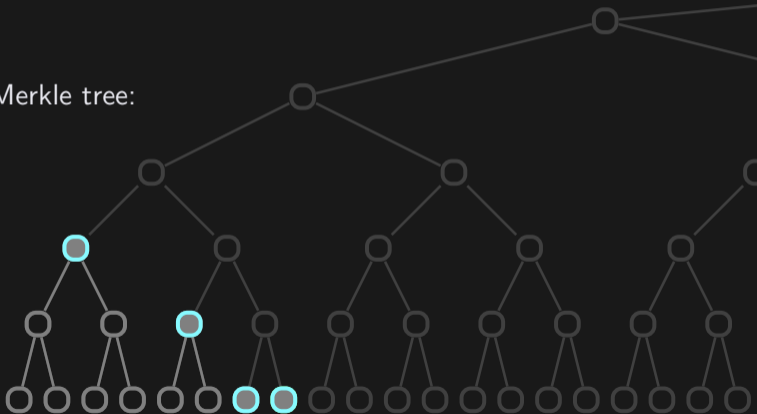


Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible

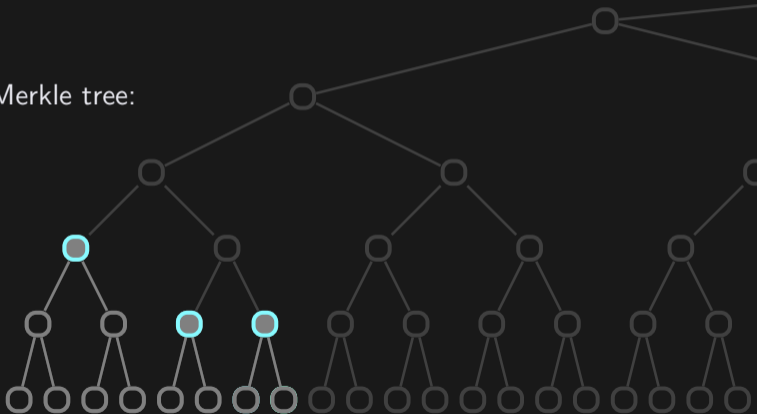


Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible

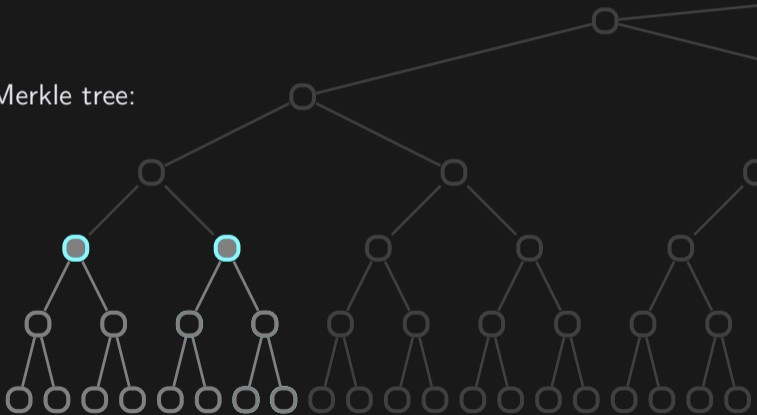


Merkle Mountain Range

= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible

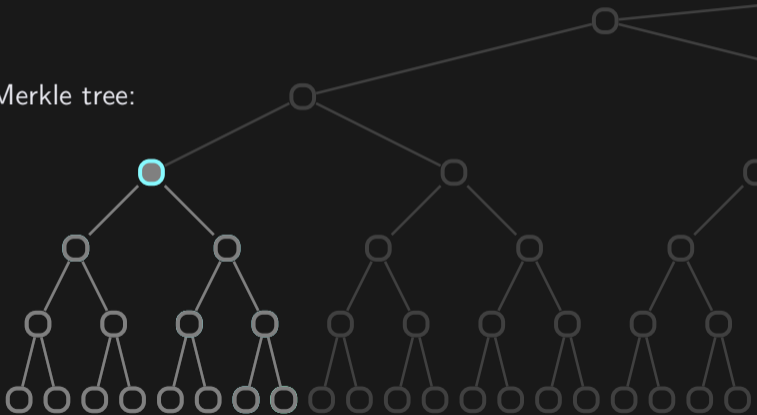


Merkle Mountain Range

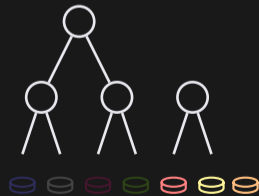
= sequence of progressively smaller Merkle trees

streaming construction of Merkle tree:

- store peaks
- append leafs
- merge and forget
whenever possible



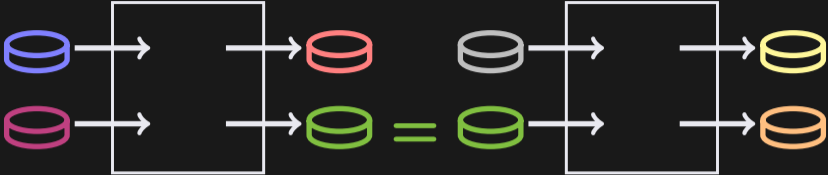
Succinct Blockchain with MMRs



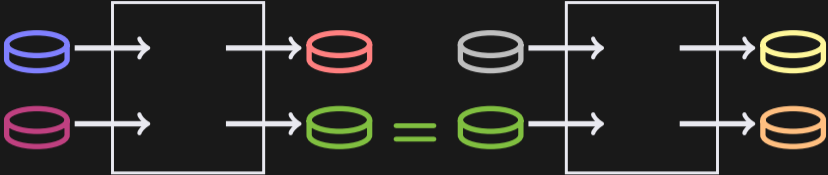
Privacy Problem



Privacy Problem



Privacy Problem

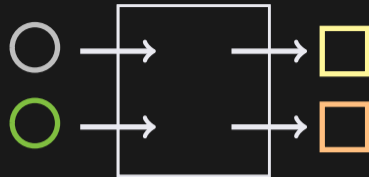
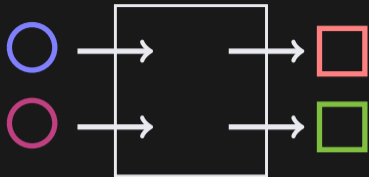


transparent ledgers: auditable ✓ private ✗

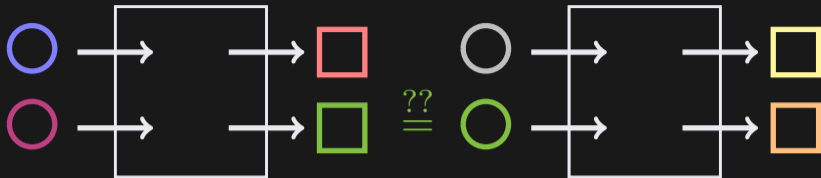
opaque + zkps: auditable ✓ private ✓

Ideal Construction

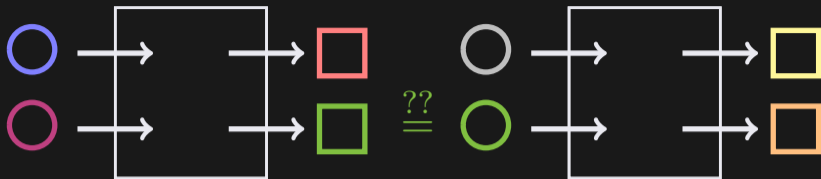
Ideal Construction



Ideal Construction

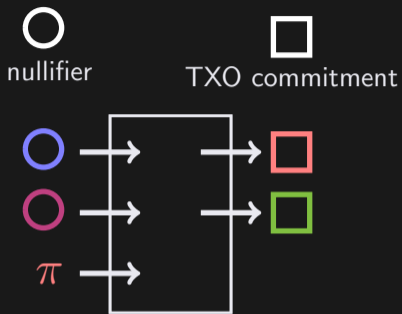


Ideal Construction

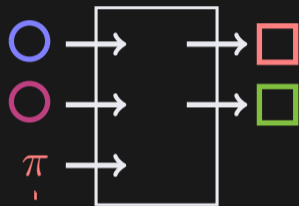
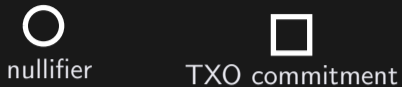


Addition records (\square) and removal records (\circ) are *distinct* and *unlinkable* cryptographic commitments to the *same* UTXO.

Decoy and Nullifier Sets

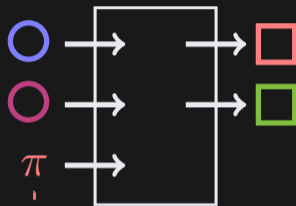
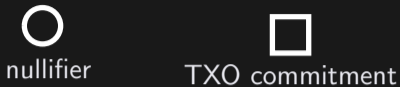


Decoy and Nullifier Sets



$\exists \square \mapsto \circ$ and \circ is new

Decoy and Nullifier Sets



$\exists \square \mapsto \circ$ and \circ is new

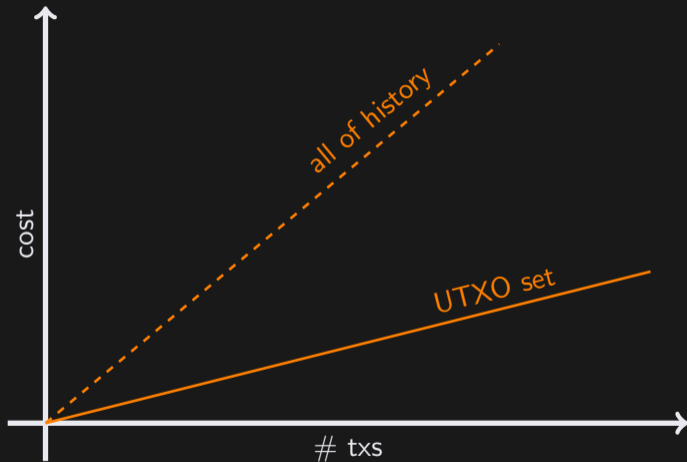


⇒ not scalable

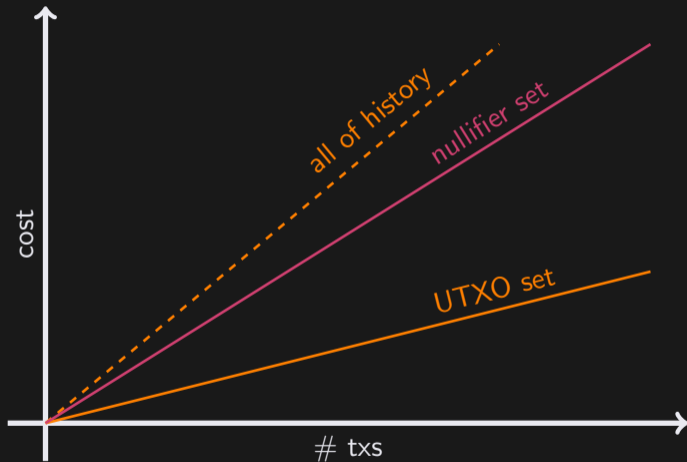
Scalability Problem



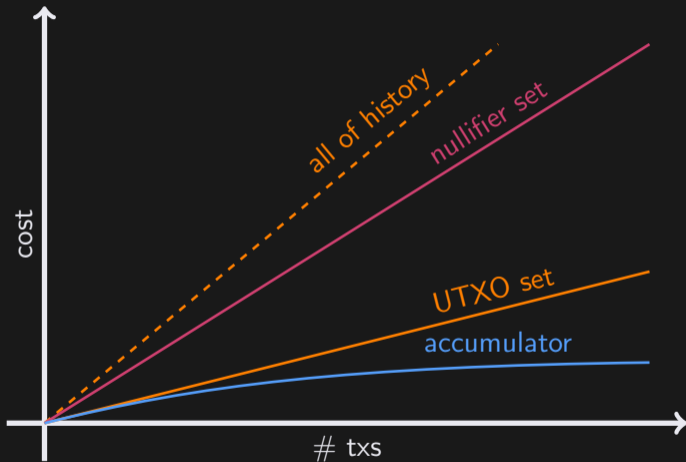
Scalability Problem



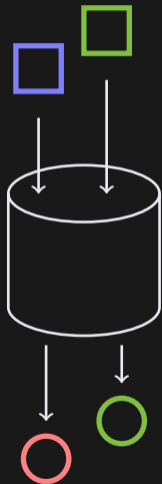
Scalability Problem



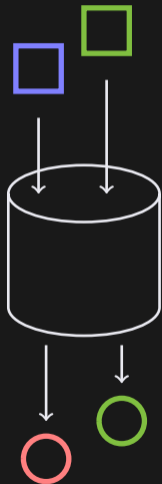
Scalability Problem



Mutator Set



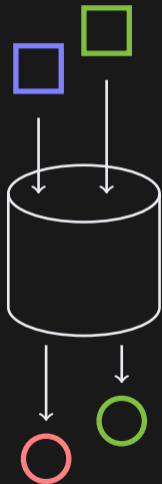
Mutator Set



A *Mutator Set* is a cryptographically authenticated data structure satisfying:

- ✓ put items in \rightarrow *addition record*
- ✓ take items out \rightarrow *removal record*
- ✗ inspect items
- ✗ remove non-members
- ✗ link removals to additions
- *scalable*

Mutator Set

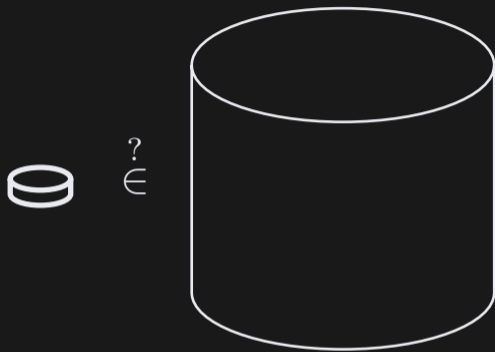


A *Mutator Set* is a cryptographically authenticated data structure satisfying:

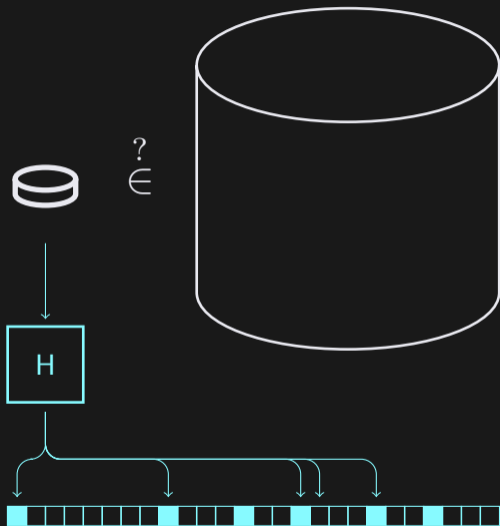
- ✓ put items in \rightarrow *addition record*
- ✓ take items out \rightarrow *removal record*
- ✗ inspect items
- ✗ remove non-members
- ✗ link removals to additions
- *scalable*
 - add: $O(1)$
 - remove: $\tilde{O}(\log N)$
 - update: $O((\log N)^2)$

Bloom Filter

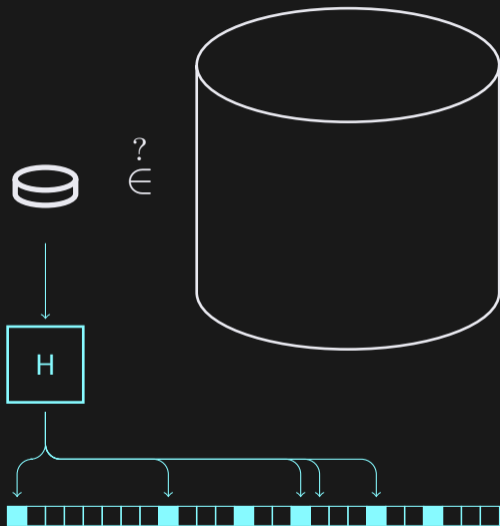
Bloom Filter



Bloom Filter



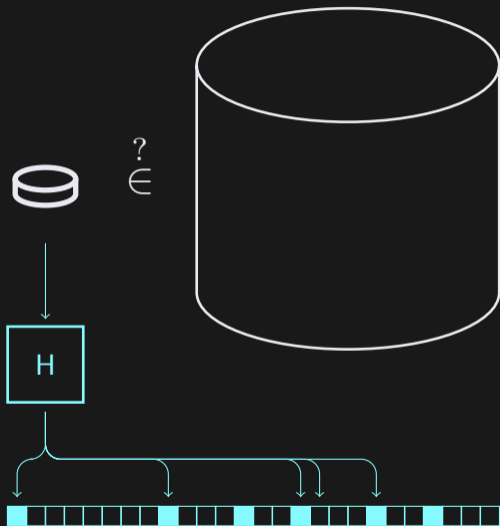
Bloom Filter



$$\Pr[\text{false neg.}] = 0$$

$$\Pr[\text{false pos.}] \approx (1 - e^{kn/w})^k$$

Bloom Filter



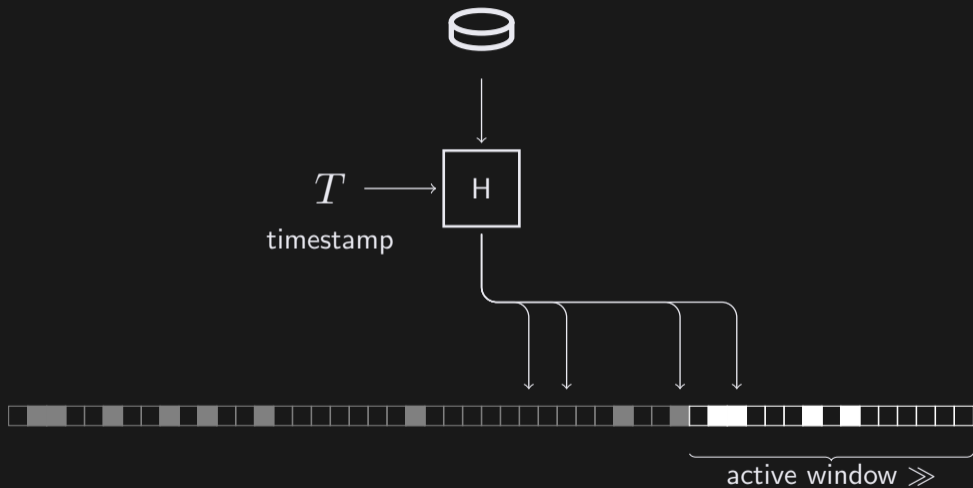
$$\Pr[\text{false neg.}] = 0$$

$$\Pr[\text{false pos.}] \approx (1 - e^{-kn/w})^k$$

✓ can be *negligible*

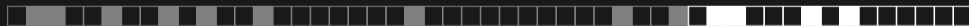
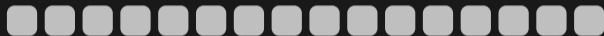
✗ finite capacity

Sliding Window Bloom Filter



Basic Construction

append-only commitment list (AOCL)

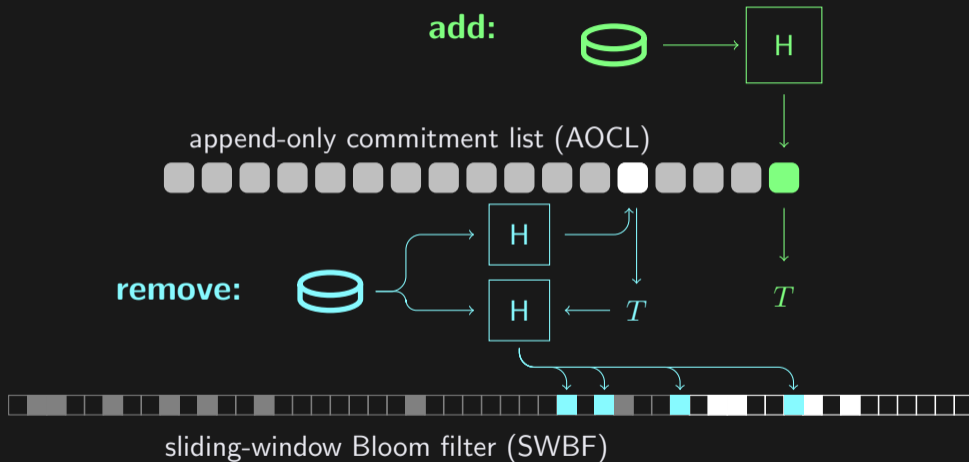


sliding-window Bloom filter (SWBF)

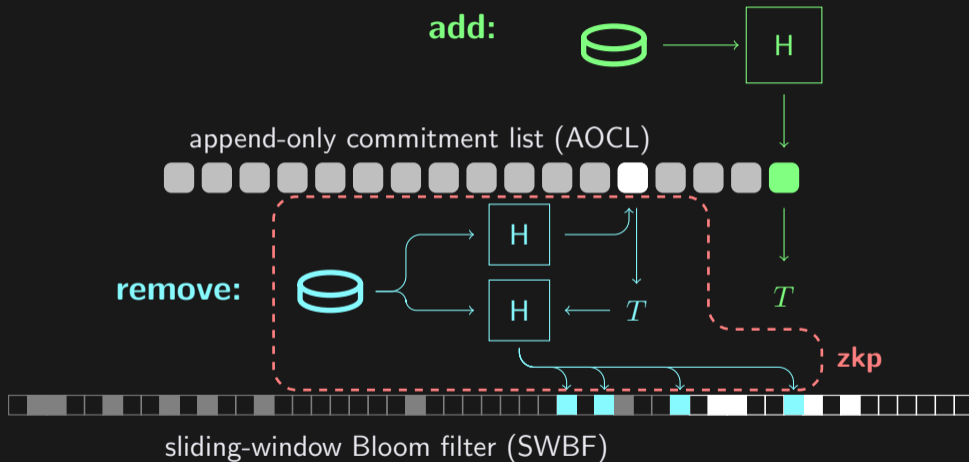
Basic Construction



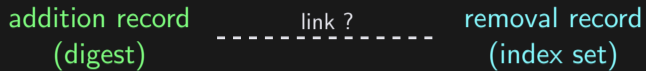
Basic Construction



Basic Construction



Privacy



Privacy



Privacy

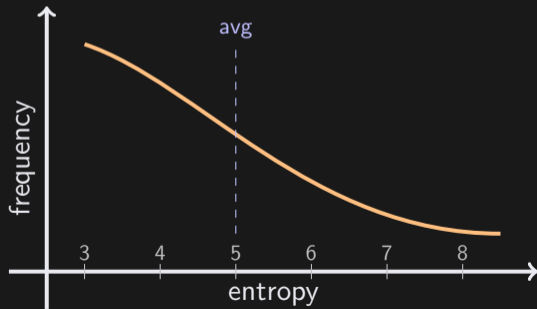
addition record
(digest)

link ?

removal record
(index set)

distribution of
plausible origins

← fuzzy timestamp



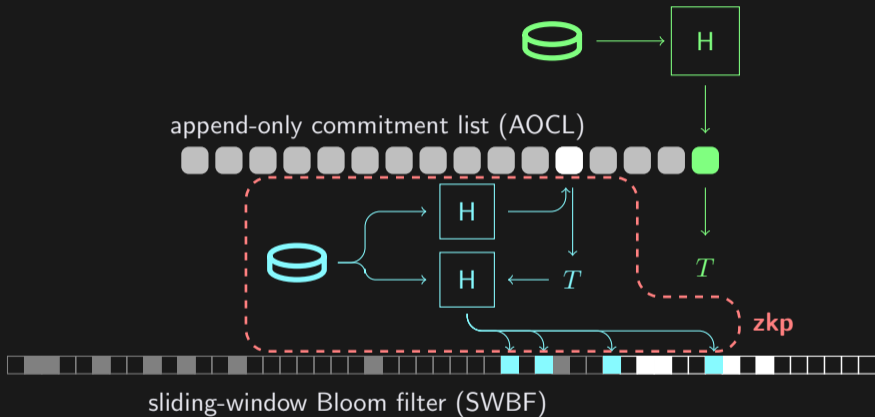
$$w = 2^{20}$$

$$b = 8$$

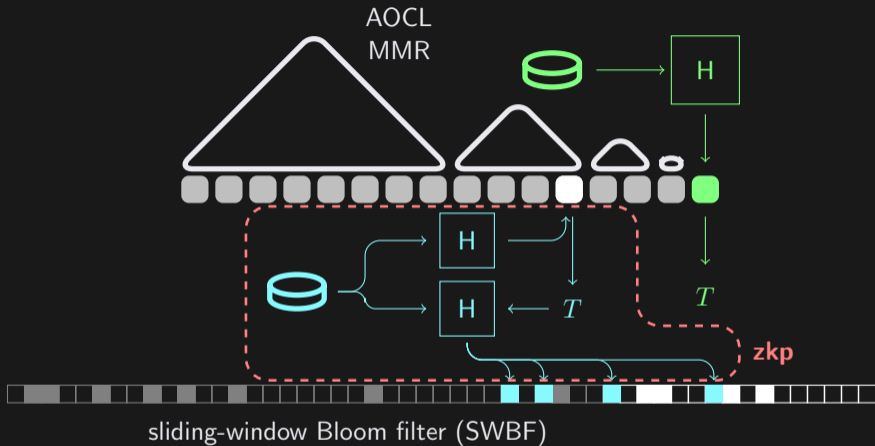
$$s = 2^{12}$$

$$k = 45$$

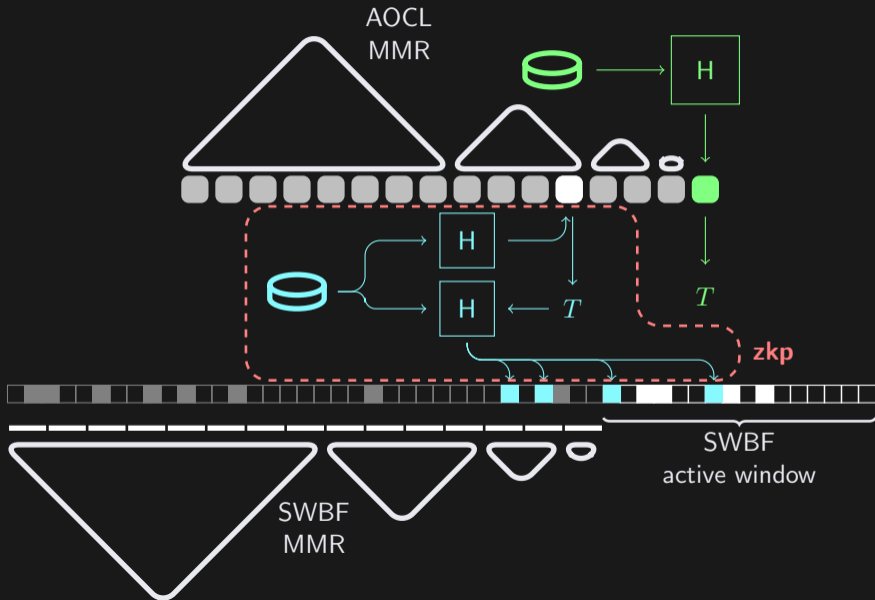
Mutator Set



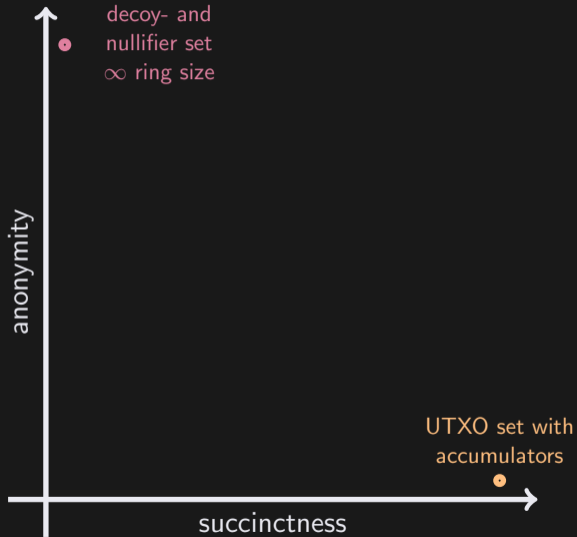
Mutator Set



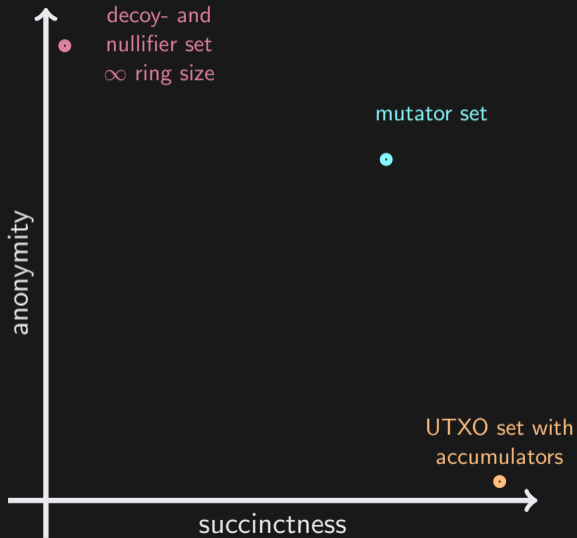
Mutator Set



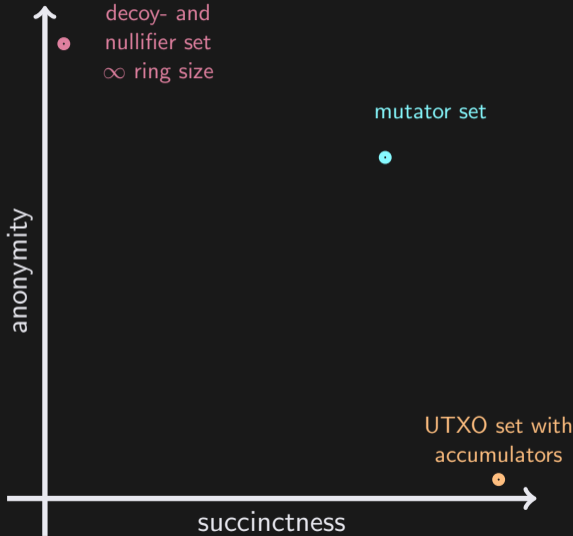
Tradeoff (?)



Tradeoff (?)

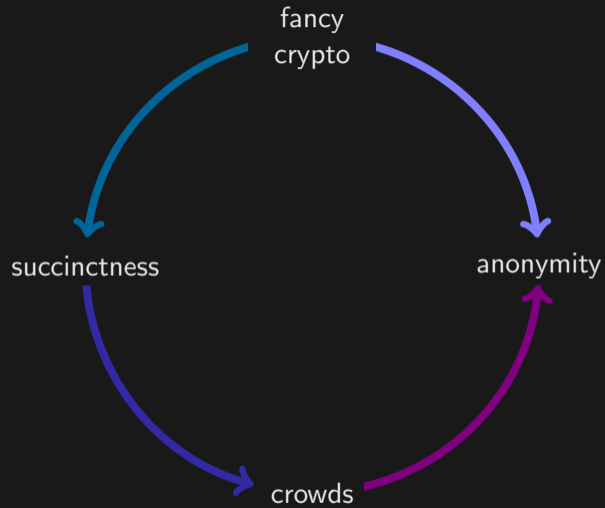


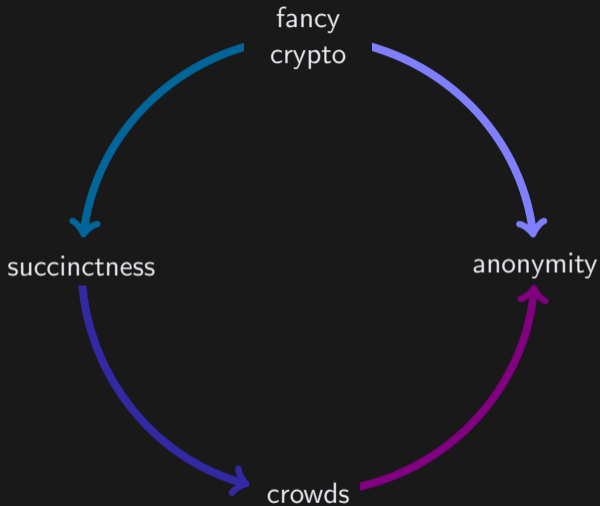
Tradeoff (?)



$$\langle a \cdot s \rangle > 0$$

metrics are
not orthogonal





speaker: Alan Szepieniec

艾伦·余丕涅茨

alan@neptune.cash