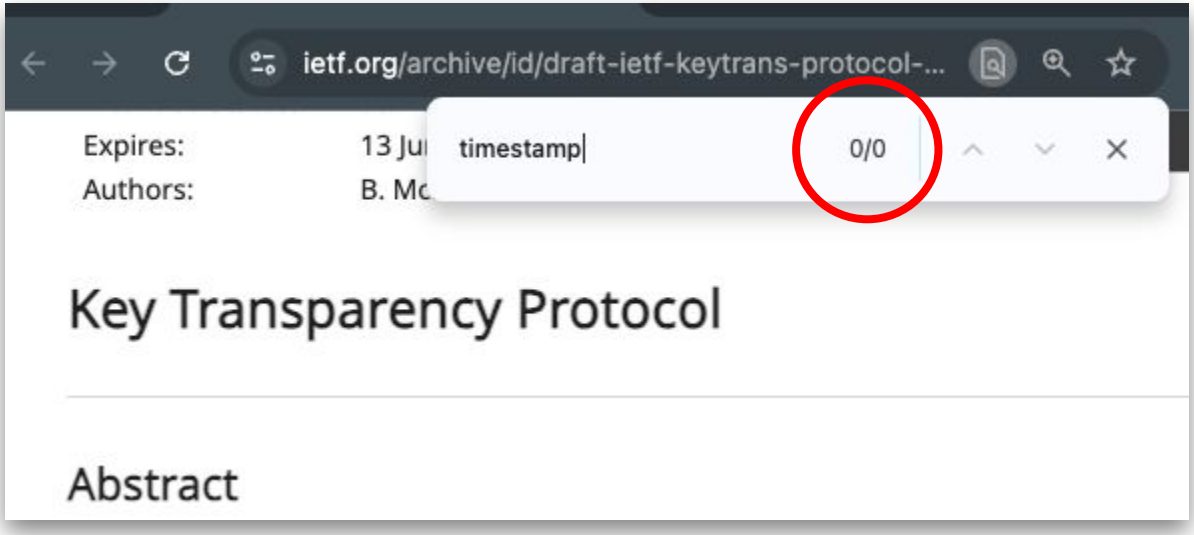
draft-ietf-keytrans-protocol

Brendan McMillion
IETF 122 / March 17, 2025

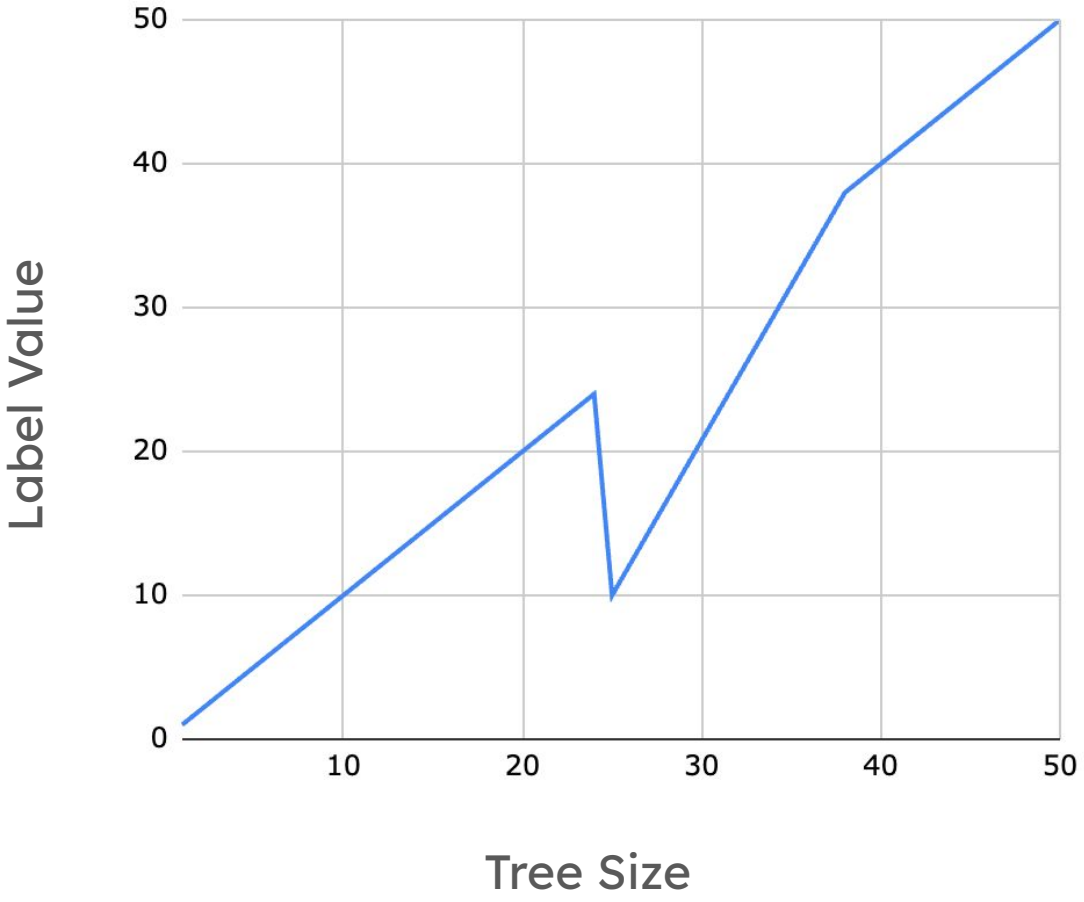
Chapter 1:
Authenticating Time

First draft had no concept of time:



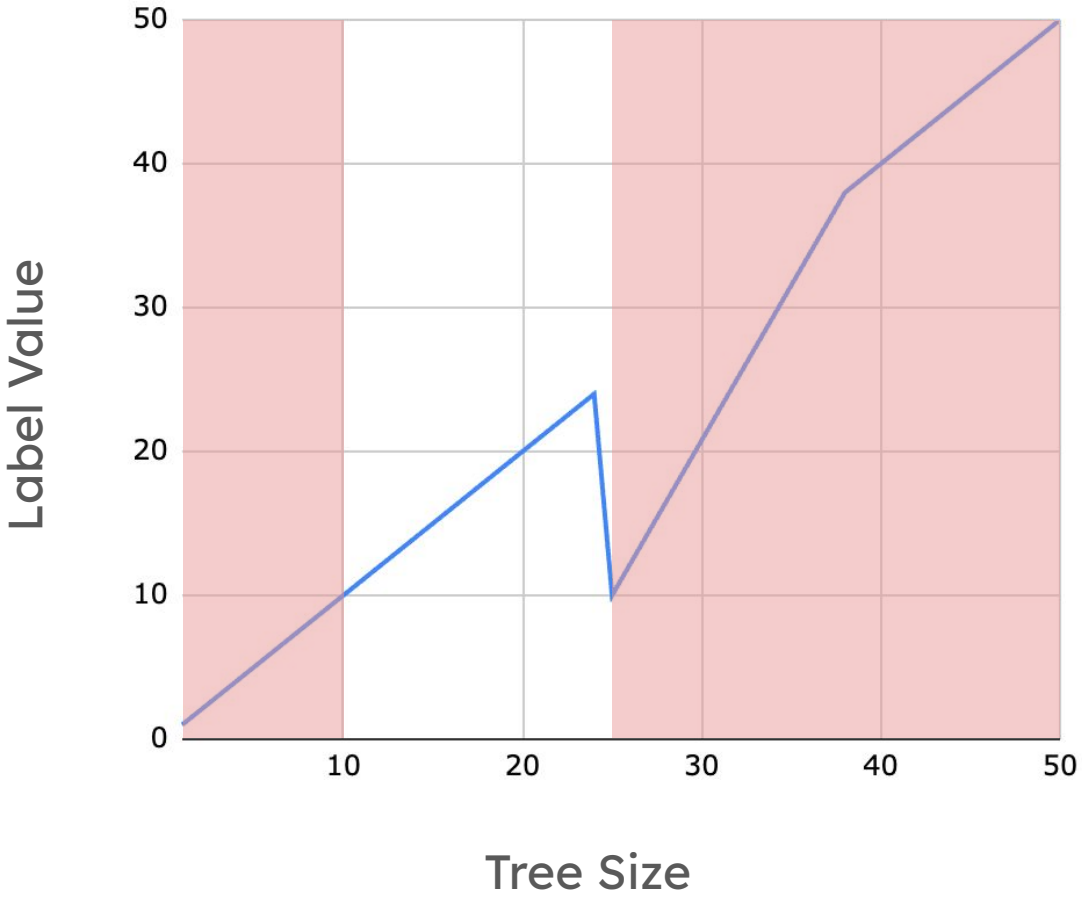
Thought was: Surely, applications can handle this on their own?

Clock Labels



Clock Labels

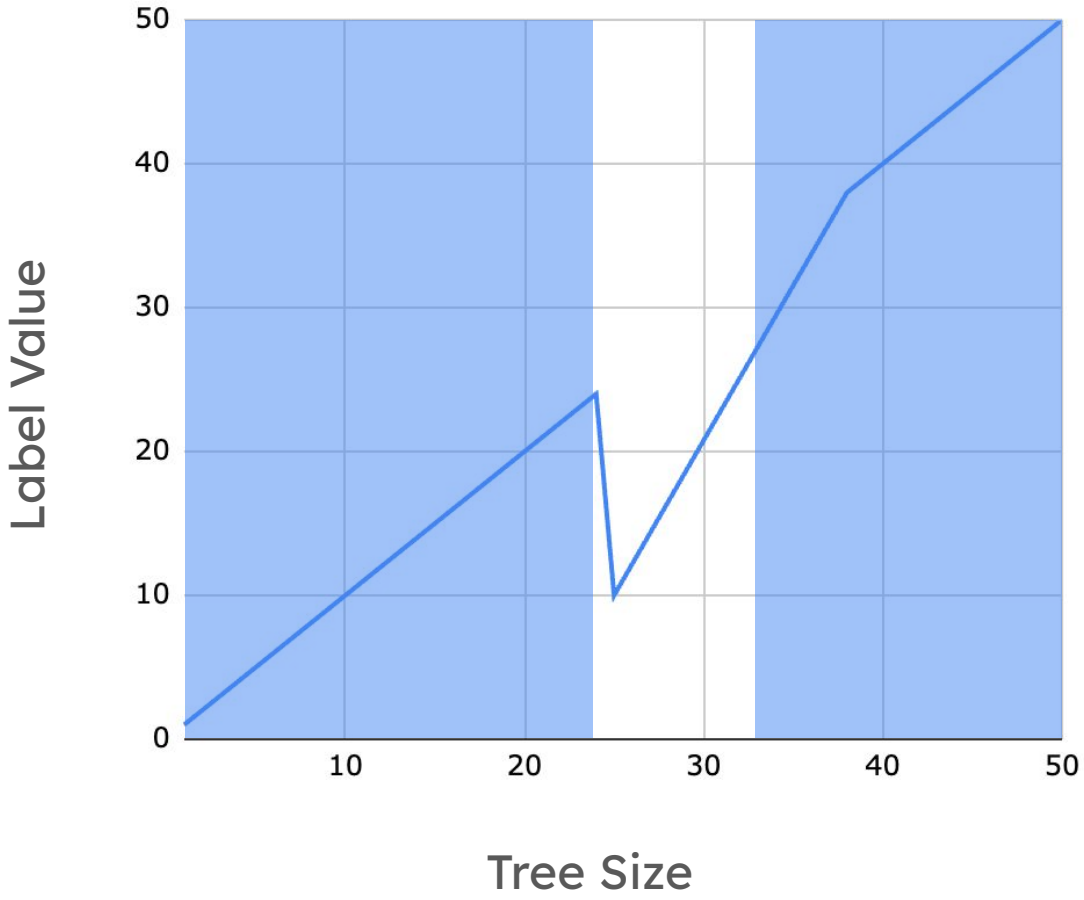
If the Log only shows Alice tree heads in the shaded region, she'll think the label's value is monotonic



Clock Labels

If the Log only shows Bob tree heads in the shaded region…

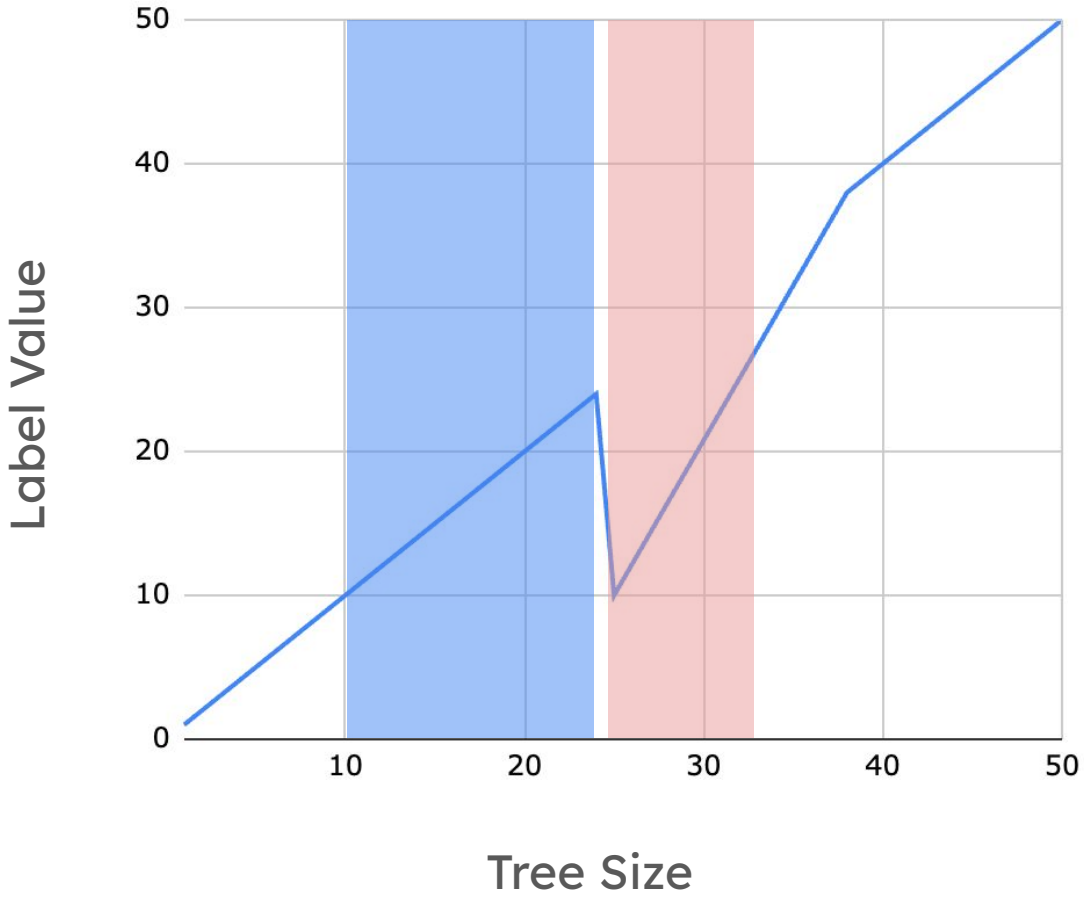
He'll also think the label's value is monotonic!



Clock Labels

Alice and Bob temporarily get put on different “forks”

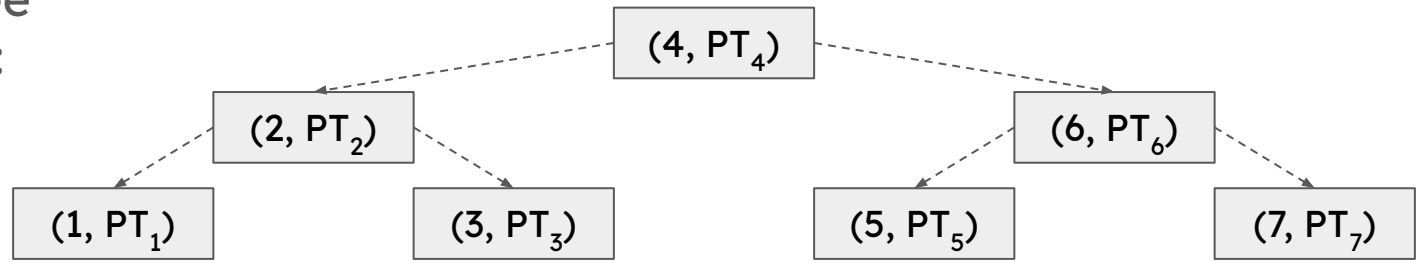
Once the attack is over, Alice and Bob are back on the same view! The attack is undetectable!



Solution: Timestamps

- Log leaves: Prefix Tree Hash => (Timestamp, Prefix Tree Hash)
- Users already navigate tree according to the Implicit Search Tree
  - Users now also check that timestamps are consistent with search tree structure

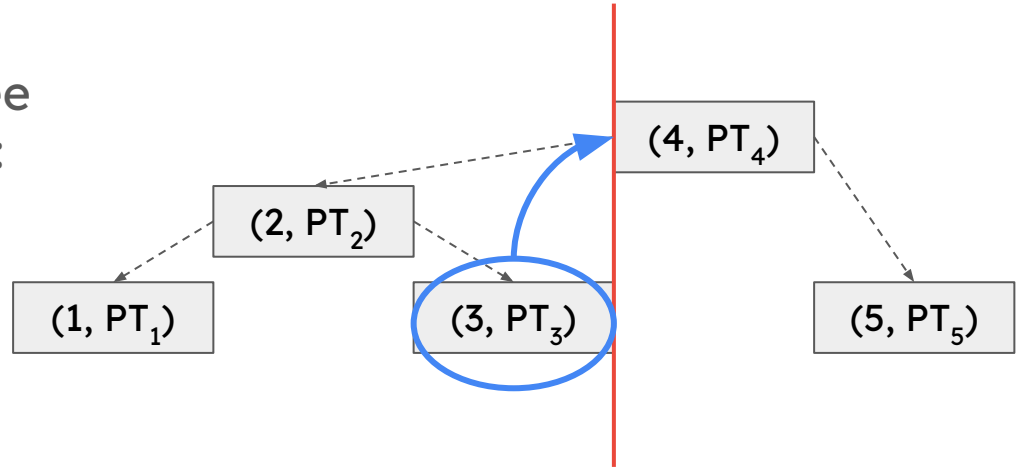
Log Tree Leaves:



Updating Tree Heads

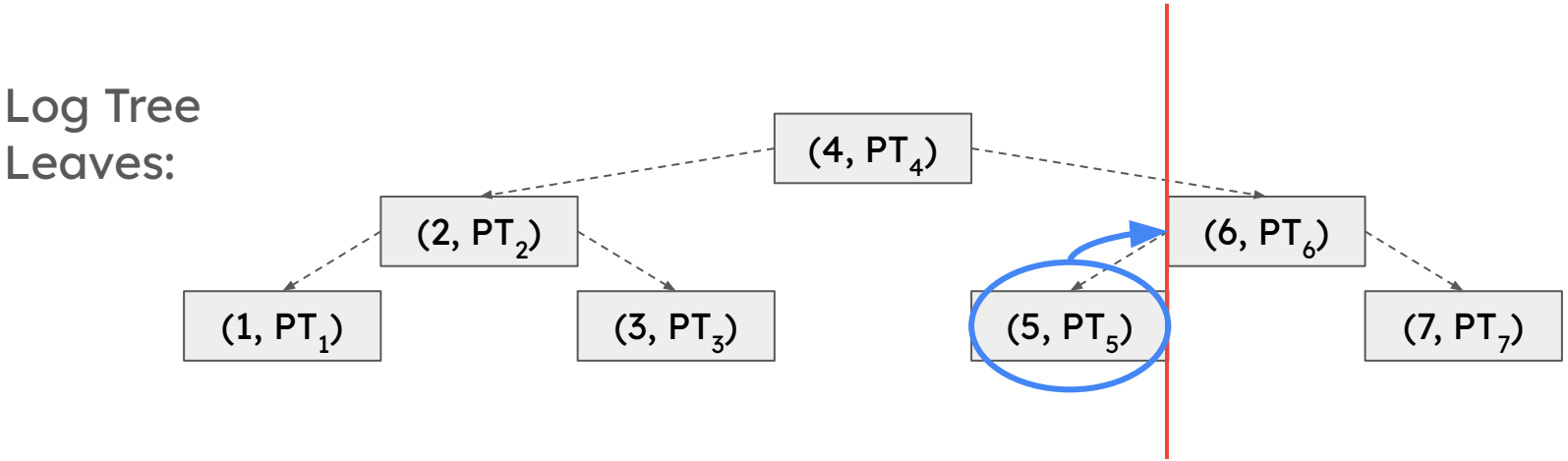
- When moving to a newer tree head, users check that previous log entries were not “covered up”

Log Tree Leaves:



Updating Tree Heads

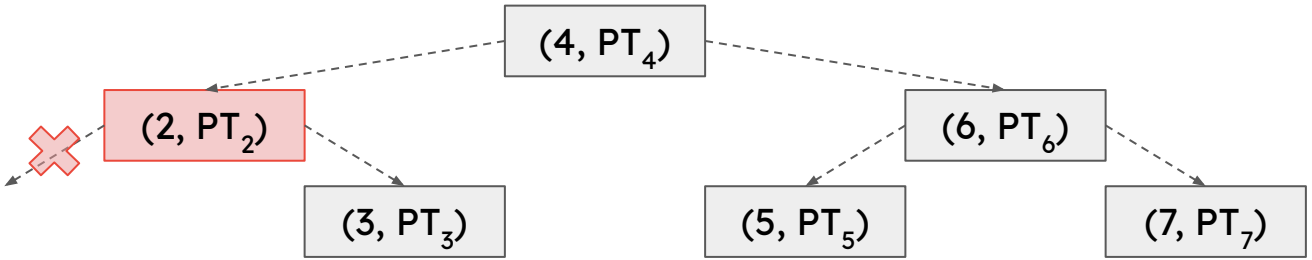
- When moving to a newer tree head, users check that previous log entries were not “covered up”



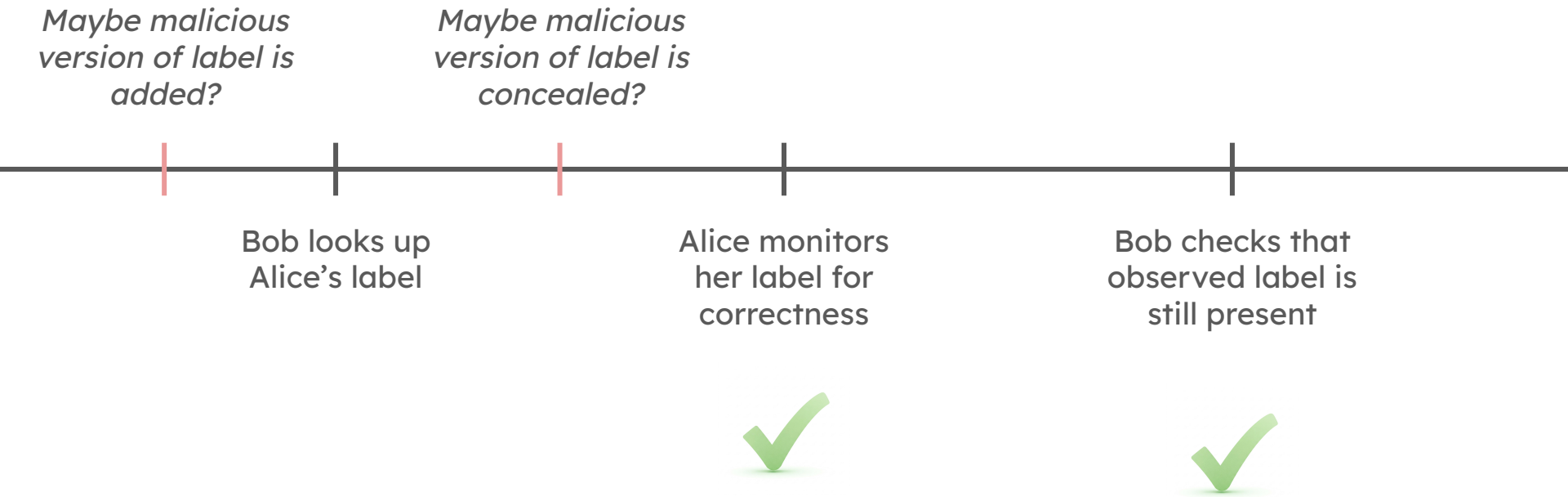
Feature #1: Maximum Lifetime

- Logs can define an amount of time after which log entries die
- Searches that hit a dead log entry can try to move back into the non-dead portion of the tree, or return an error if that's not possible
- Allows provable, secure pruning!

Log Tree Leaves:



Feature #2: Reasonable Monitoring Window



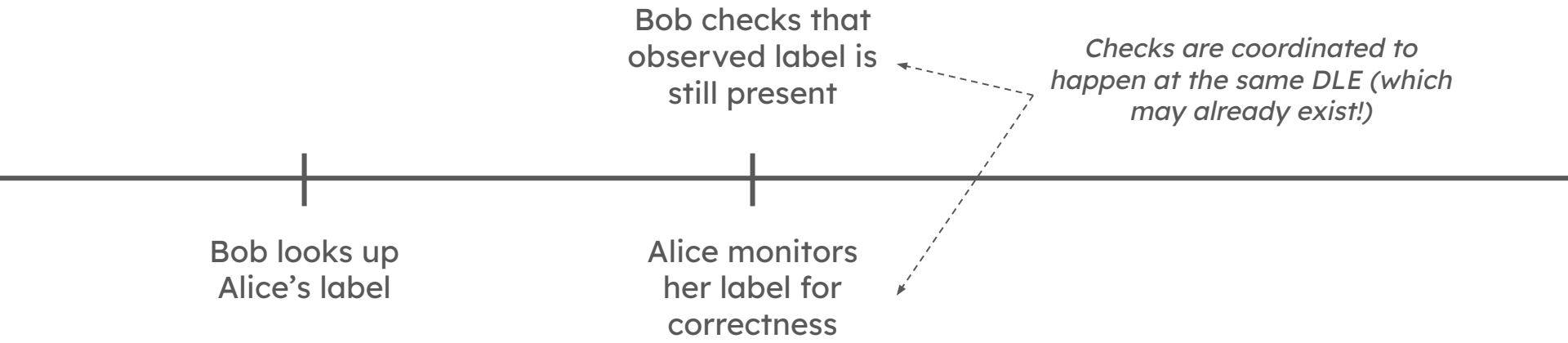
Feature #2: Reasonable Monitoring Window

- If log entries have timestamps now…
- Then when I look up a label, I can see how long ago it was inserted.
- Shouldn’t that convey something about how trustworthy the value is?

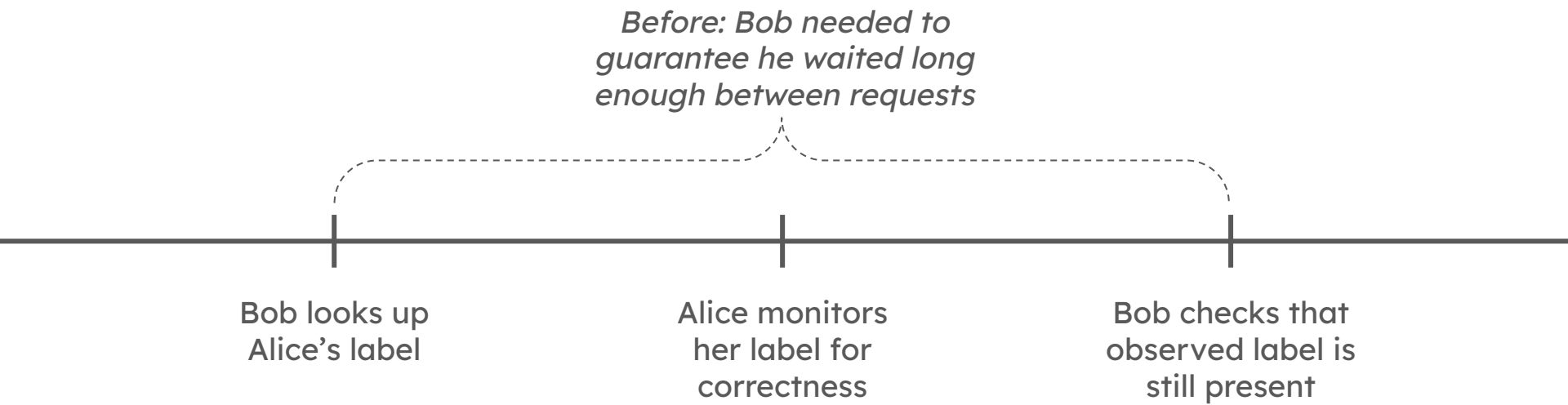
Reasonable Monitoring Window (RMW): How often the Transparency Log *generally expects* users to monitor labels they own.

Feature #3: Distinguished Log Entries

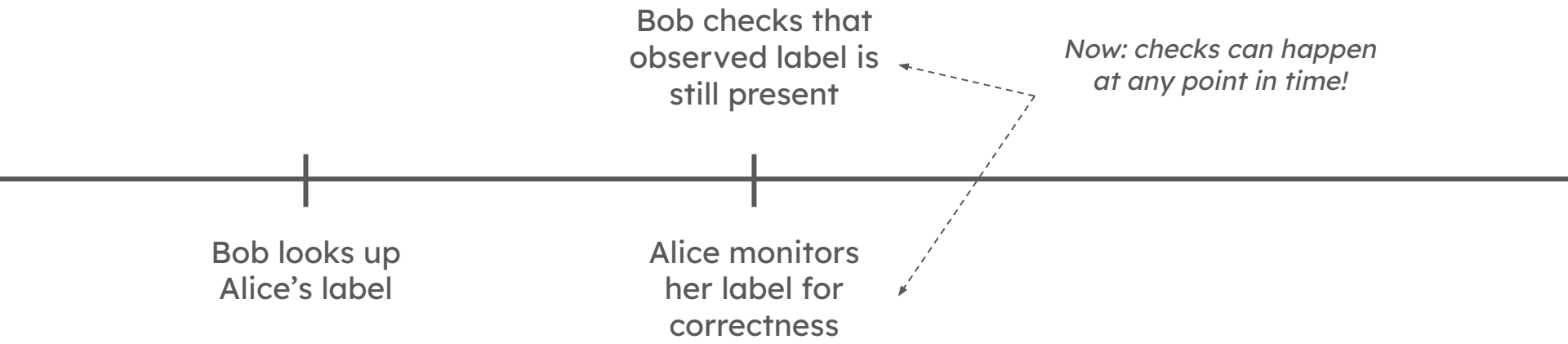
- Log entries chosen according to an algorithm with these criteria:
  - **Regularly Spaced:** Roughly one Distinguished Log Entry per interval of the RMW
  - **Stable:** Once distinguished, always distinguished
  - **Verifiable:** Everybody agrees what is/isn't distinguished



Feature #3: Distinguished Log Entries



Feature #3: Distinguished Log Entries



Users can be offline for a long time and "catch up"

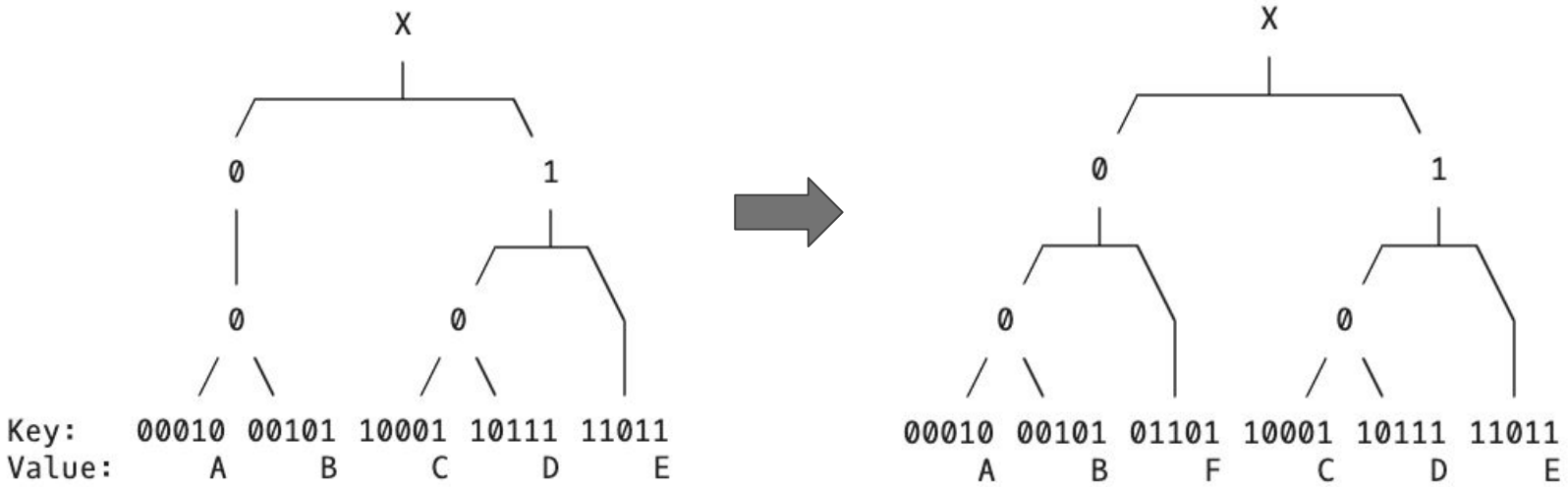
⇨

Users that look up a label can retain it for less time

Chapter 2:
Self-Balancing Prefix Tree

“Prefix Tree” – Store key-value pairs

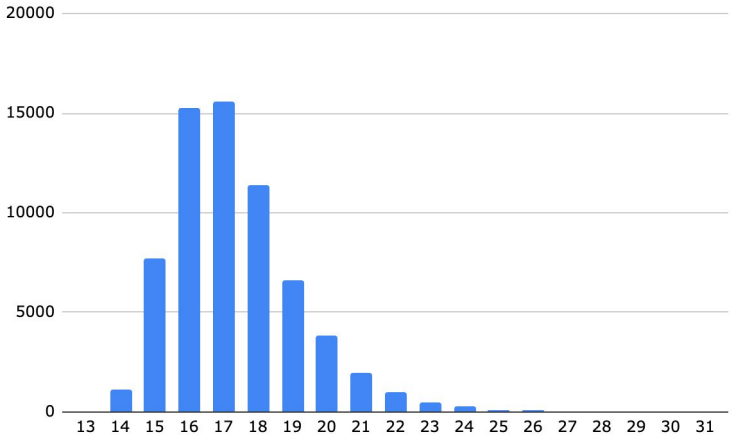
Trie: Every node represents a prefix that all strings stored below it share. The node’s left child contains all strings that share the same prefix plus a ‘0’ bit; the right, plus a ‘1’



Motivation

- Interest in using KT for non-privacy-sensitive applications (where a VRF would be wasteful)
  - Problem: Without a VRF, it's very easy to unbalance the tree

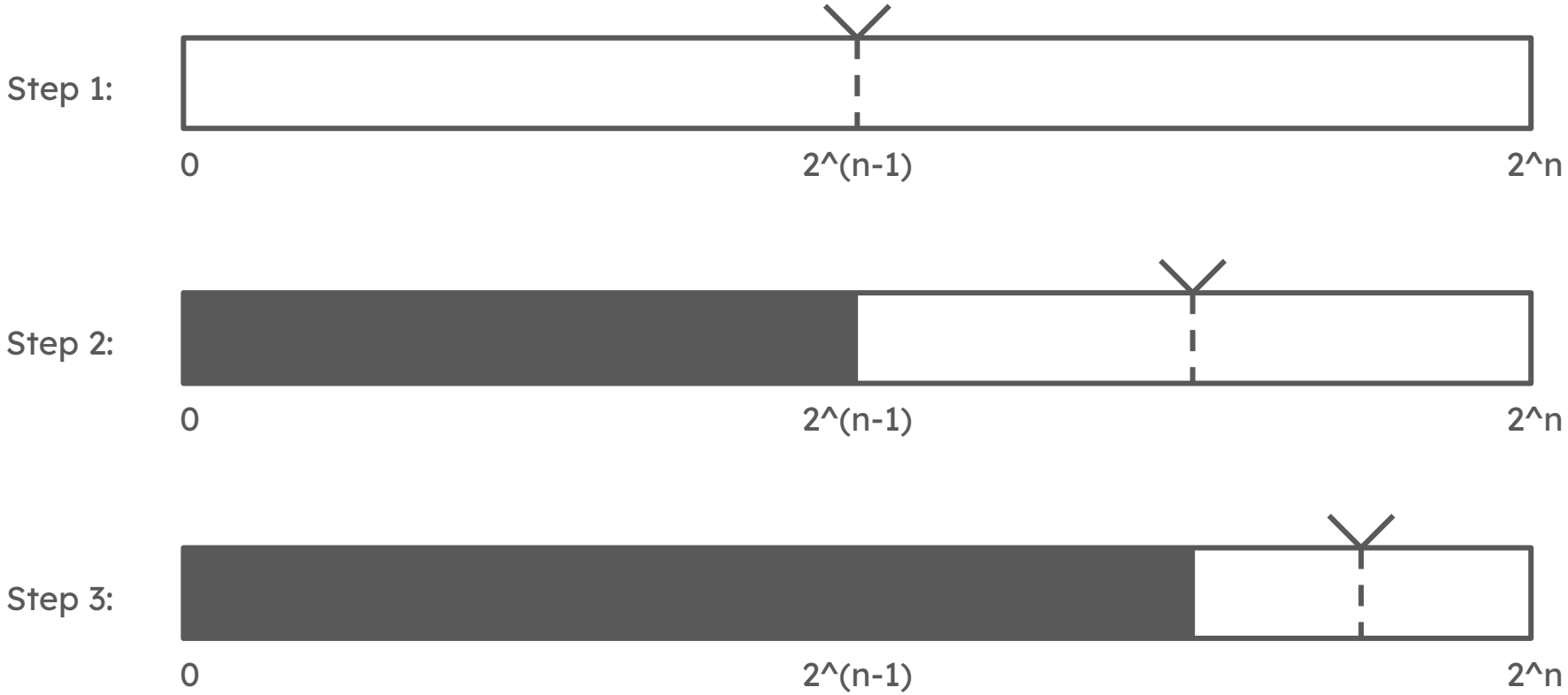
- Probabilistic balancing tends to have a very wide spread anyway:



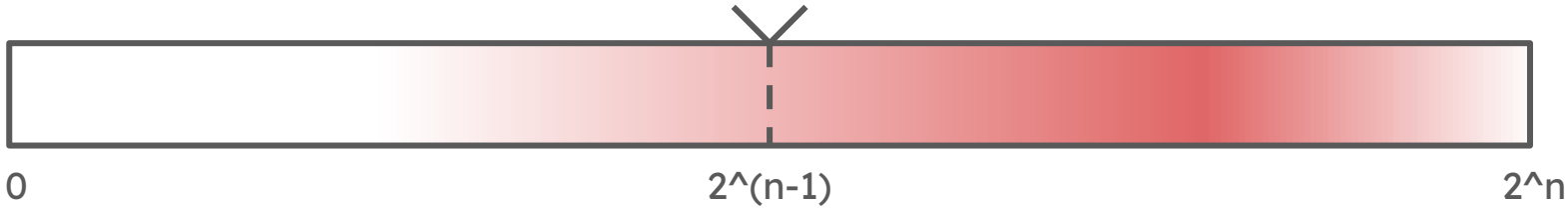
Problem

Typical Balanced Search Trees store data in their intermediate nodes:

- Rather than providing just copath hashes, we'd need to provide copath hashes **and** each node's stored data in cryptographic proofs
- Would make proofs twice as large :(

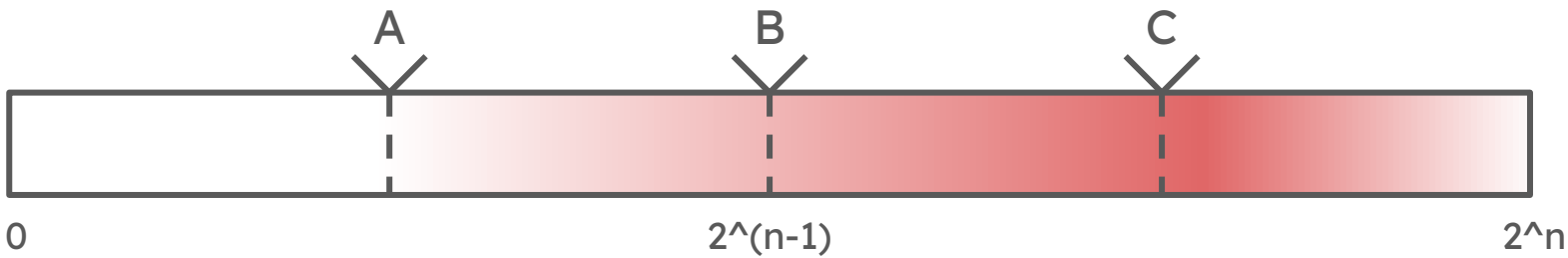

Step 1:
0
2^(n-1)
2^n
Step 2:
0
2^(n-1)
2^n
Step 3:
0
2^(n-1)
2^n

Search length depends on data density, not range



Handling Density

- Allow multiple starting points in a range
- Transparency Log indicates which is most efficient



Empirical Results: Tree with 2^16 leaves

# Cursors	Avg. Leaf Depth	Exp. Deepest Leaf	Avg. Overhead
1 (Unbalanced)	17.3	34	0 bytes
3	16.8	28	4 bytes
15	16.4	23	8 bytes
255	16.2	19	16 bytes

= 2 bits for each of 16.8 search steps

Empirical Results: Tree with 2^16 leaves

# Cursors	Avg. Leaf Depth	Exp. Deepest Leaf	Avg. Overhead	Worst Case Leaf Depth
1 (Unbalanced)	17.3	34	0 bytes	256
3	16.8	28	4 bytes	~192
15	16.4	23	8 bytes	~128
255	16.2	19	16 bytes	~64

Search Length