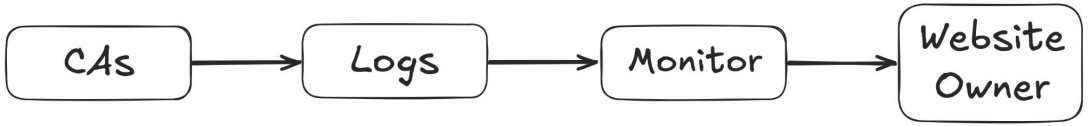
The Design Space between CT and KT

Dennis Jackson

djackson@

moz://a

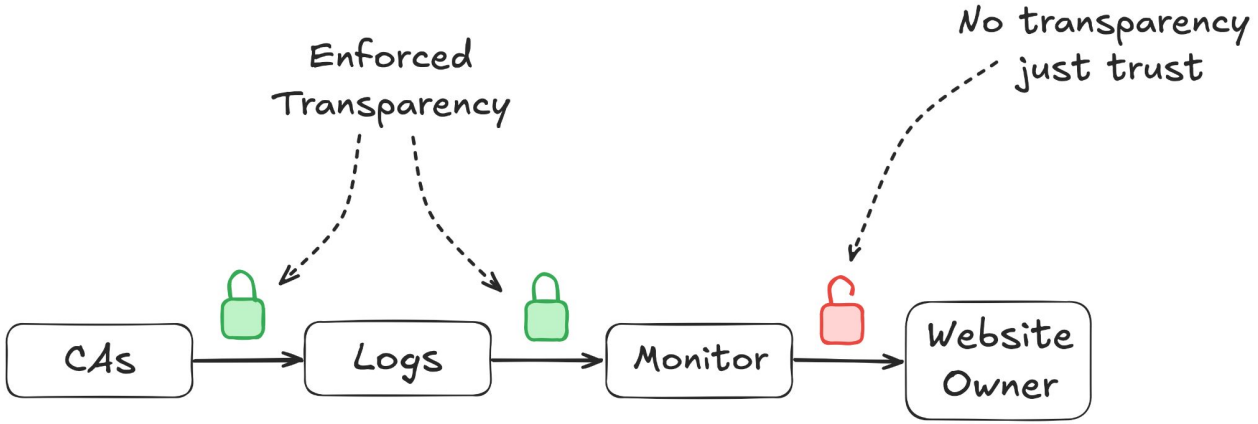
Certificate Transparency



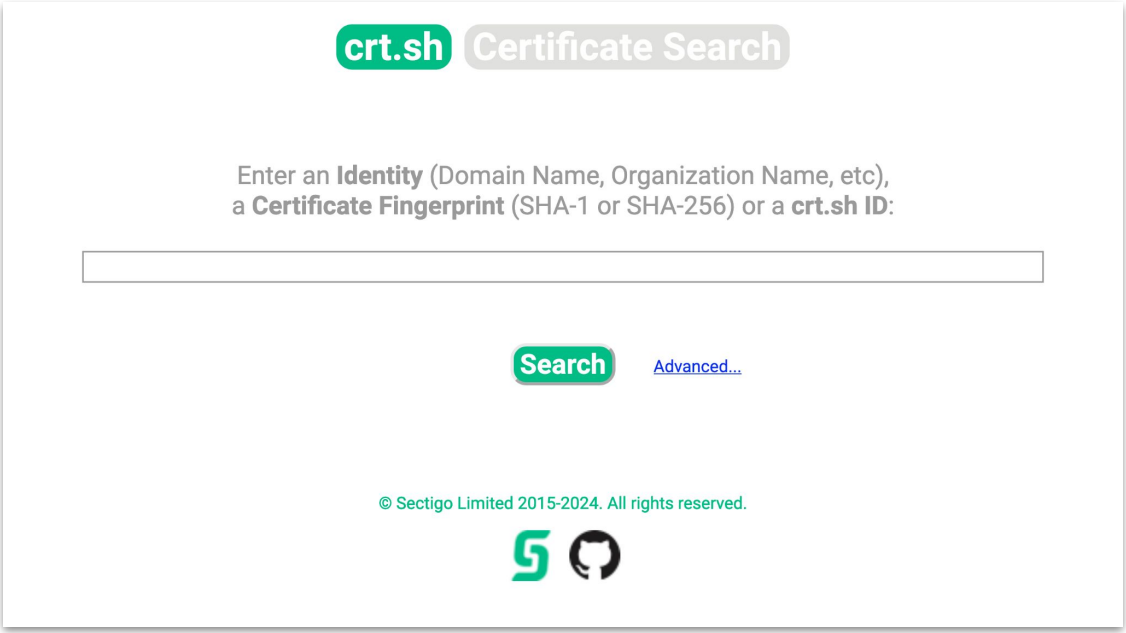
djackson@

moz://a

Certificate Transparency: Pain Points



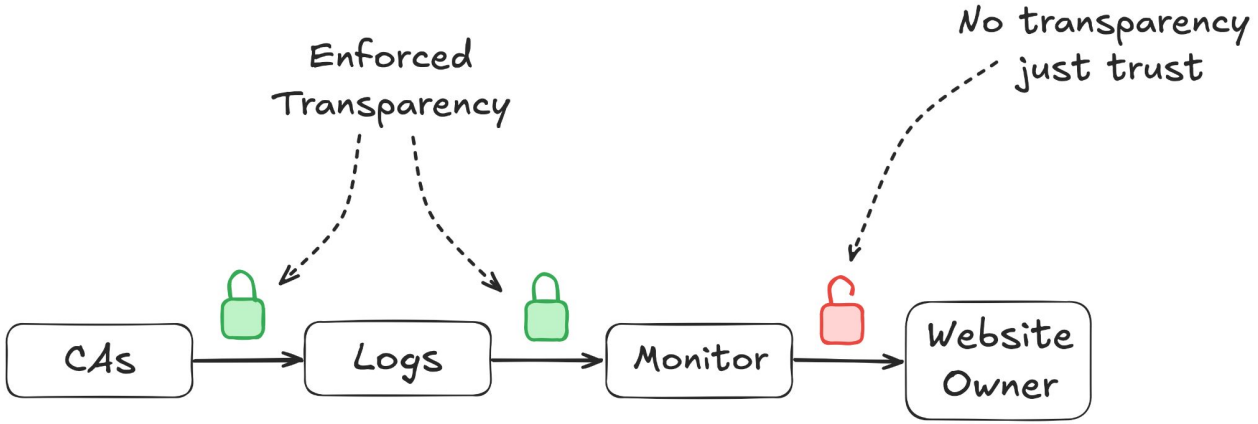
Certificate Transparency: Pain Points



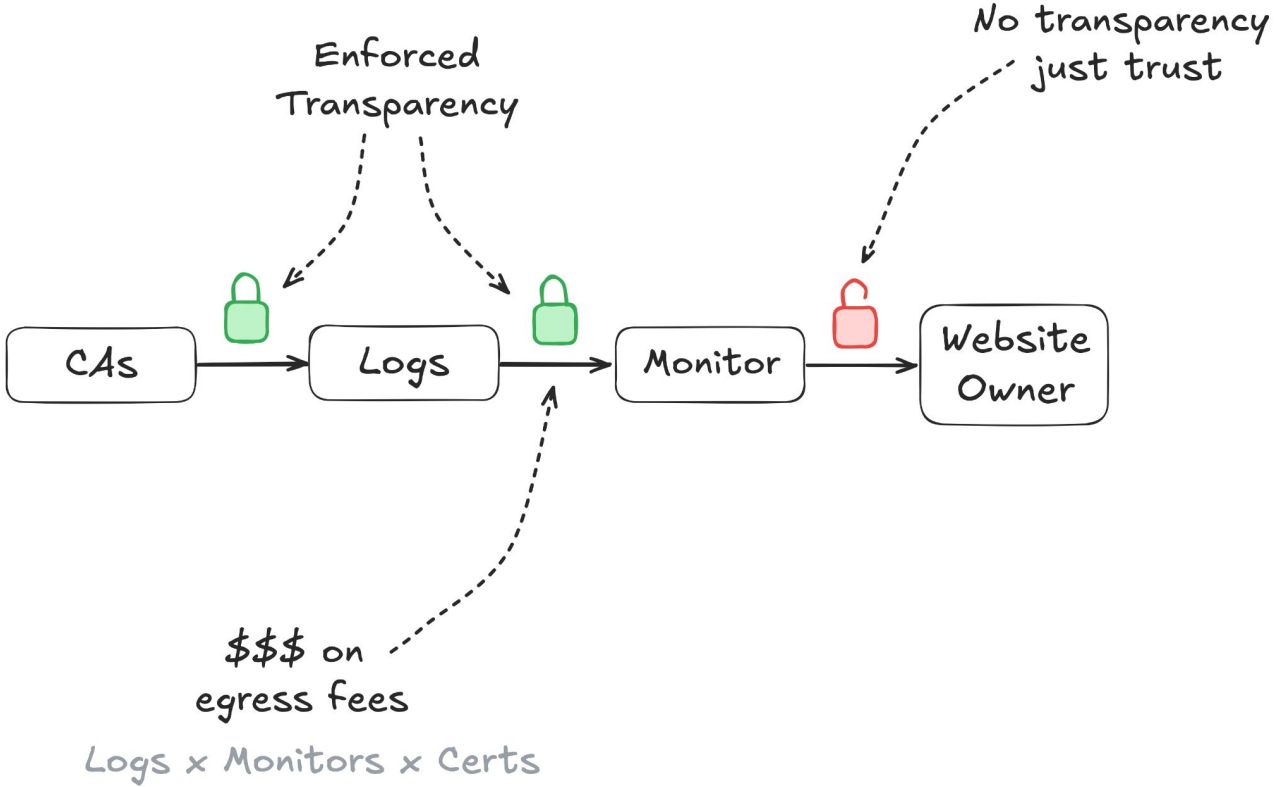
djackson@

moz://a

Certificate Transparency: Pain Points



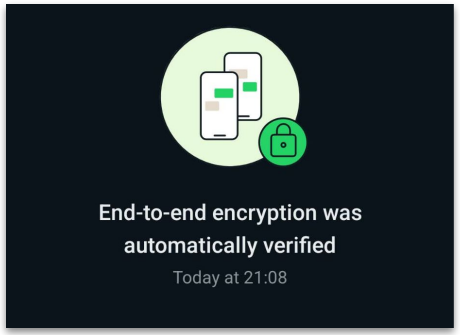
Certificate Transparency: Pain Points

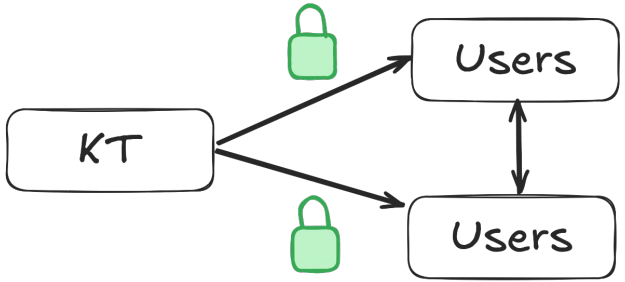


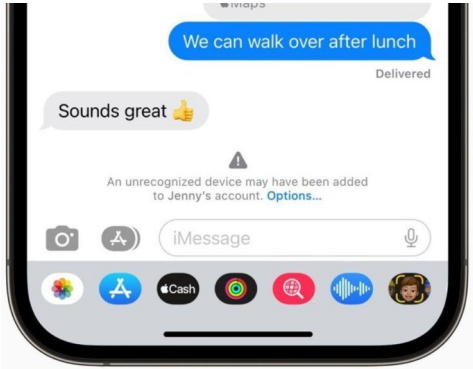
djackson@

moz://a

Key Transparency







djackson@

moz://a

The Space Between

Initial Question:

What ideas could be borrowed from KT and leveraged in a future CT system?

Observation:

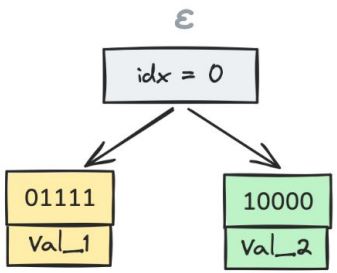
Websites, unlike users, don't need privacy.

The next 7 minutes: Sketching a design that sits between CT and KT.

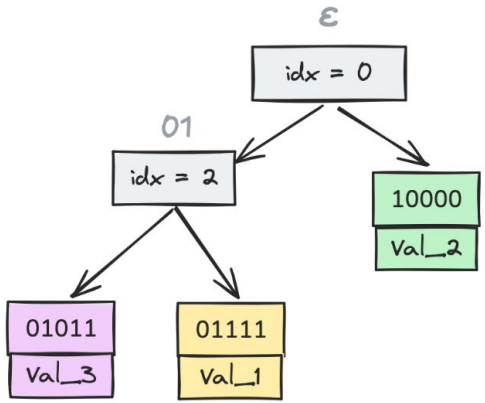
djackson@

moz://a

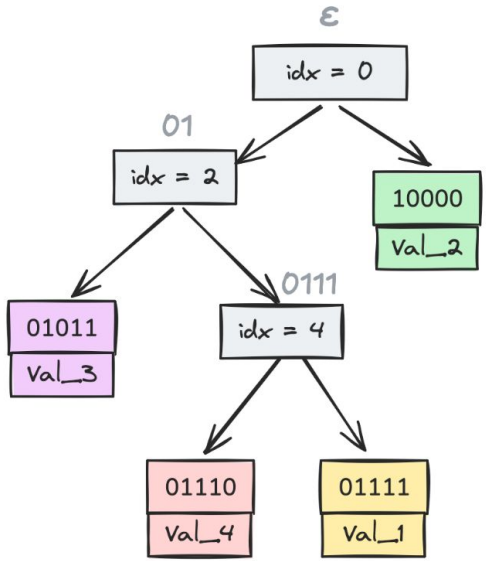
Efficient Verifiable Maps: Merkle Patricia Tries




```
{
    01111   : Val_1,
    10000 : Val_2
}
```




```
{
    01111   : Val_1,
    10000 : Val_2,
    01011   : Val_3
}
```




```
{
    01111   : Val_1,
    10000 : Val_2,
    01011   : Val_3,
    01110   : Val_4
}
```

djackson@

moz://a

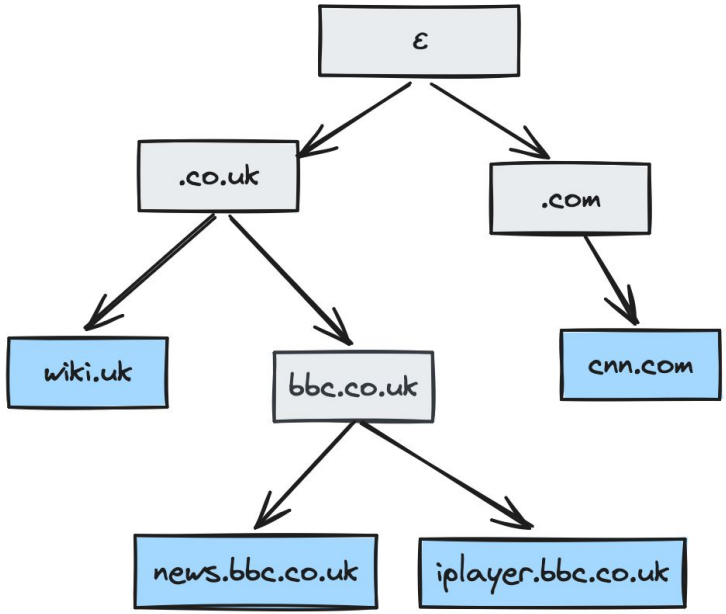
How do we key our MPT?

Key := Reverse Domain Name Notation

KT designs traditionally use a VRF to obscure user identities but we can use structured keys.

Benefits:

- Path length to root is proportional to number of eTLD+1s
- Subdomains have a shared path to the root (enables proof compression)

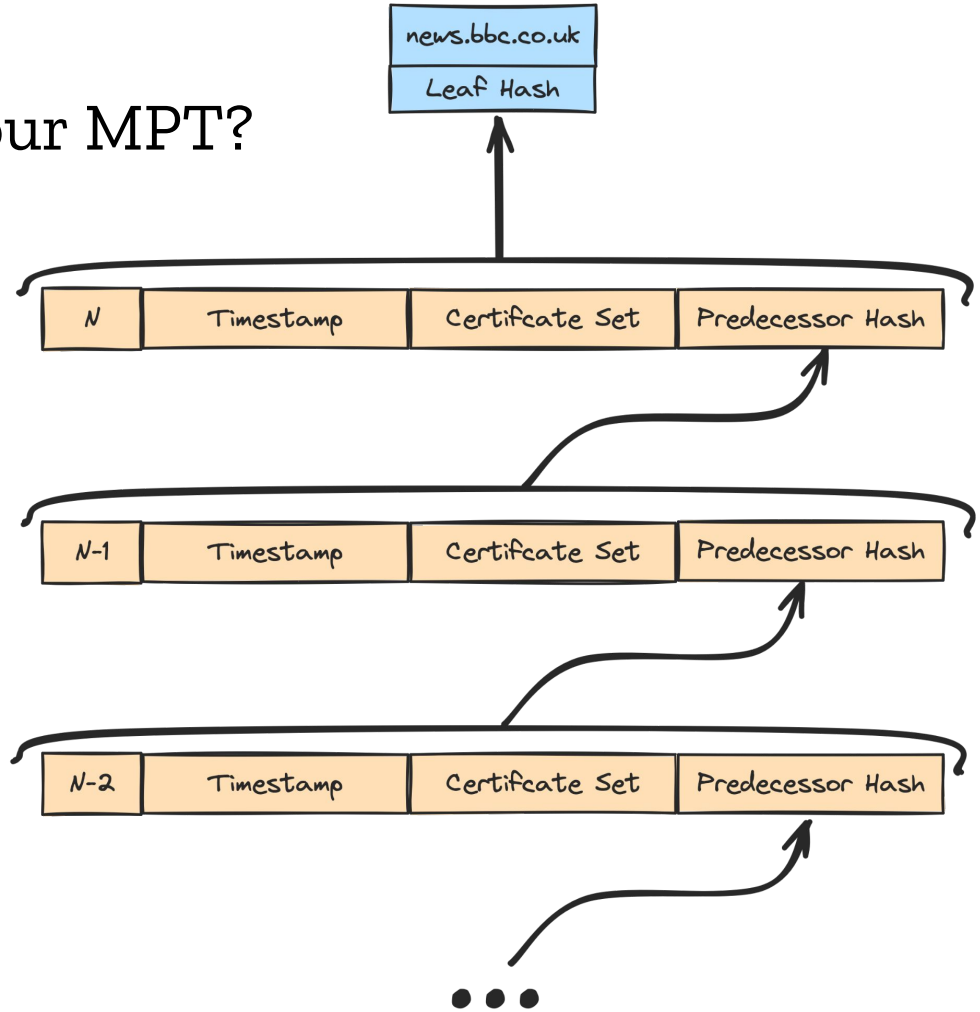


What to put in the leaves of our MPT?

A Hash Chain of Certificate Sets

Leaf Values :=

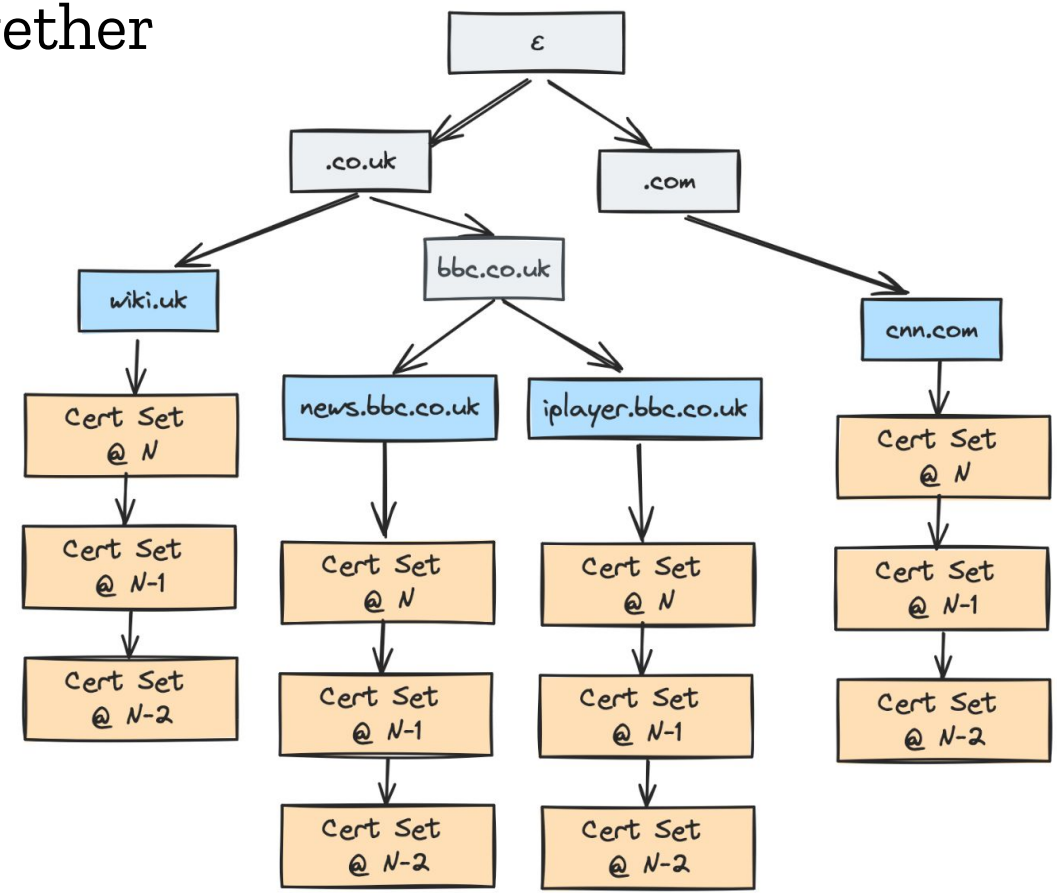
Set of Valid Certs
+ Timestamp
+ SeqNo
+ Hash of Predecessor



djackson@

moz://a

All Together

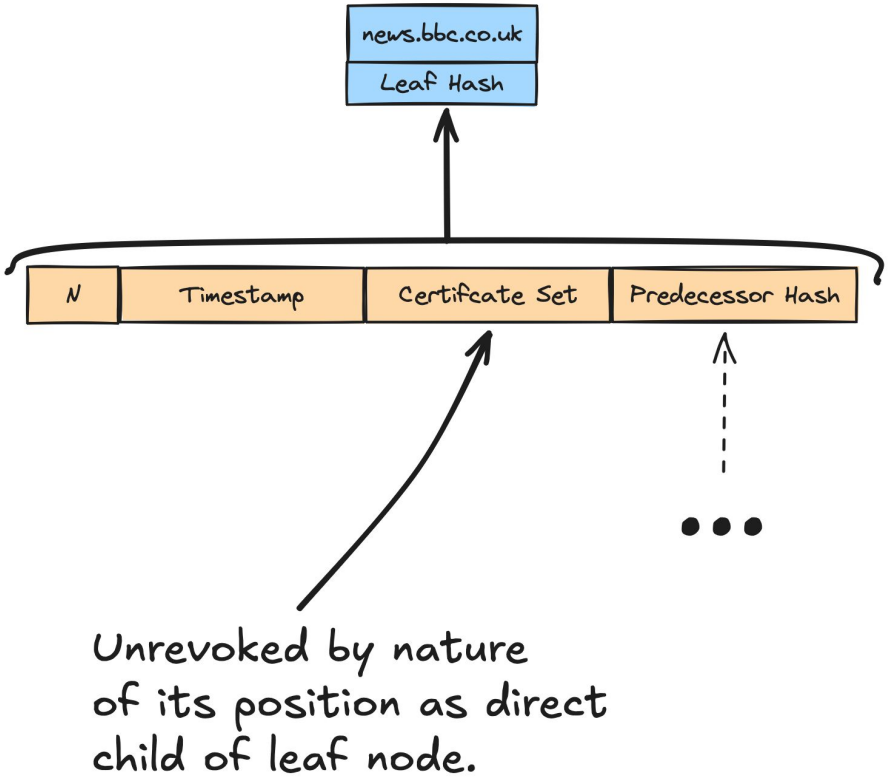


djackson@

moz://a

Revocation Transparency

- Revocation status maintained as a structural invariant.
- When a certificate gets added or revoked, we
  - Insert a new node at the top of the hash chain
  - Update its path to the root through the MPT.

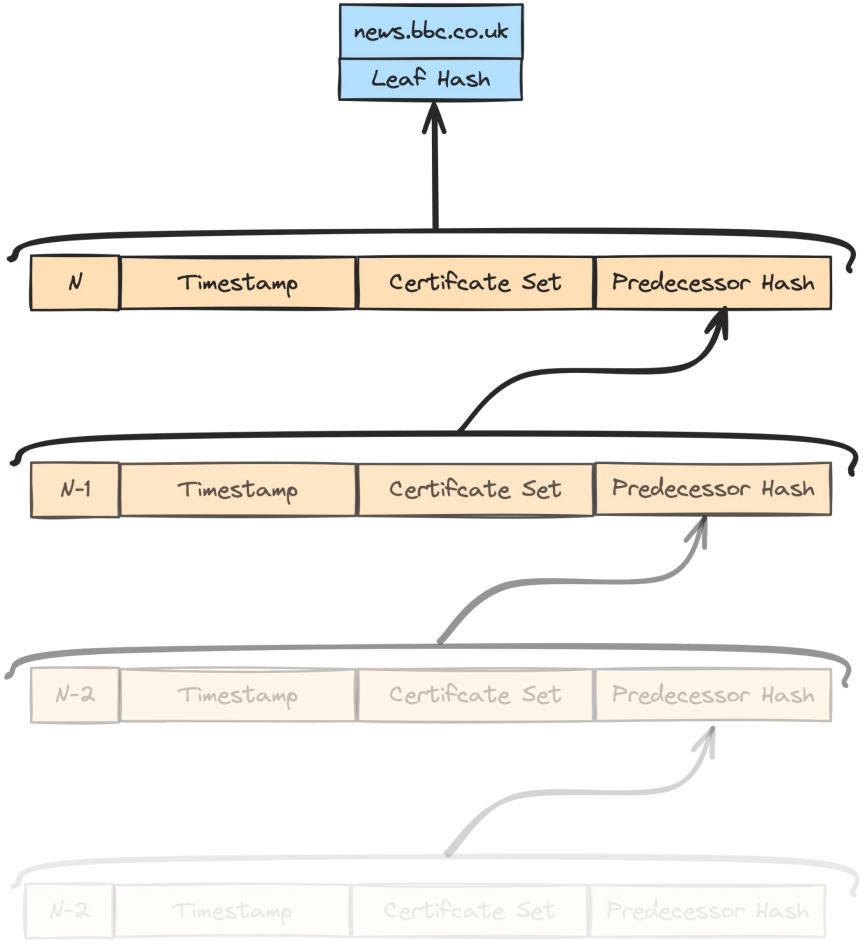


djackson@

moz://a

Graceful Expiry

- We can also gracefully forget old entries without having to update the tree.
- Log storage is proportional to number of active domains rather than total number of issued certificates.



djackson@

moz://a

Proof Lengths

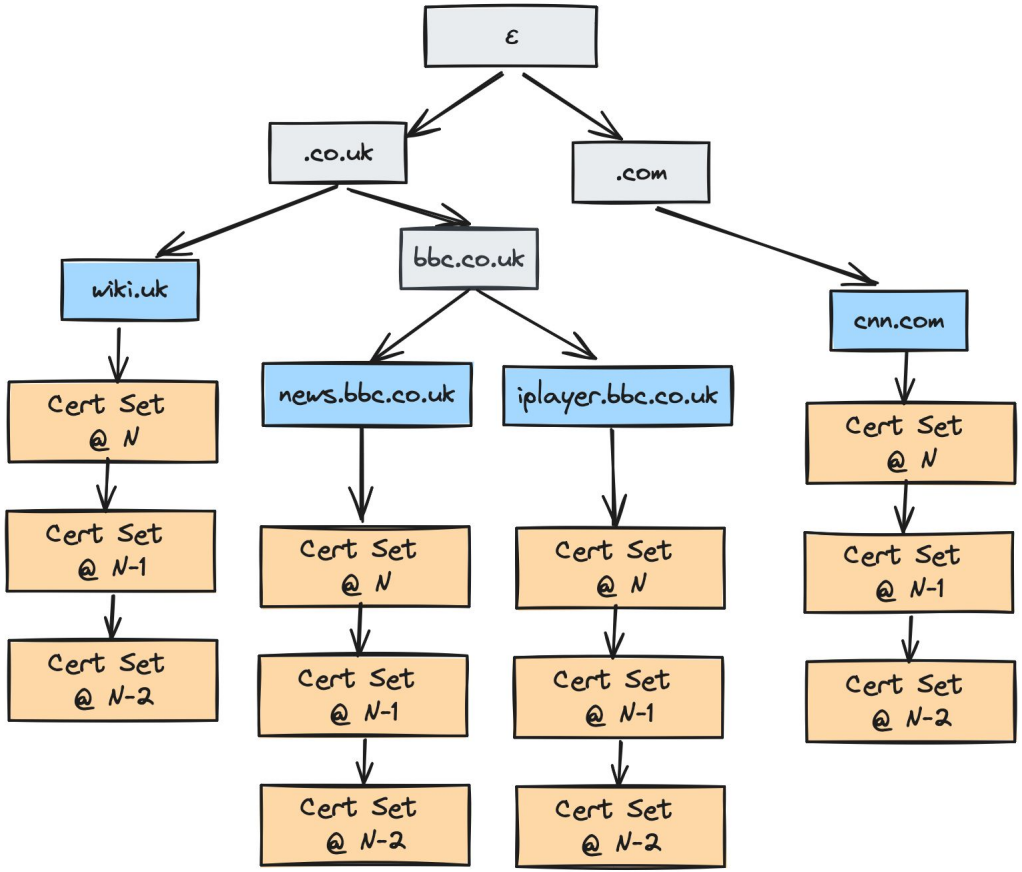
- **D** := | All eTLD+1s |
- **S** := | Site's Subdomains |
- **C** := | Site's Valid Certs |
- **E** := | Site's Expired / Revoked Certs |

Proof a certificate is not revoked (or absent):
Log **D** + log **S** + log **C**

Proof of Certificate History:
Log **D** + log **S** + **C** + **E**

Privacy Preserving KT solutions:

- Tree size scales a multiple of all certificates.
- Proof lengths are 3 x Log (|Tree size|)

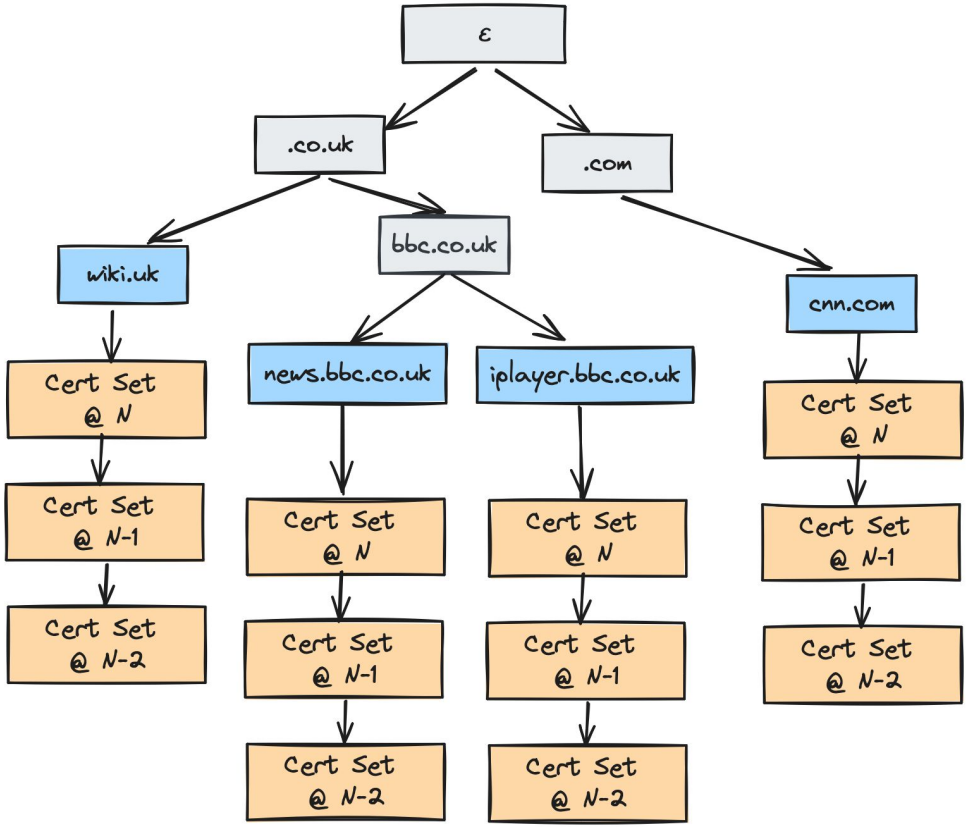


djackson@

moz://a

Observations

- Cheap E2E proof that a given certificate is present and unrevoked, e.g. in an MTC-style design.
- Enables succinct proofs of all certificates issued for a given domain and its subdomains.
- Lower egress costs - can use a quorum of auditors rather than general public monitors.
- Storage costs grow proportional to valid certificates, not total issued certificates.



djackson@

moz://a

Thoughts

- KT-like designs might help solve challenges in the CT ecosystem.
- Loosening the privacy constraint of KT unlocks a rich design space with opportunities for much greater efficiencies.
- **Ideal Monitoring Story?** `certbot --certificate-report *.example.com`
- Did this sketch pique your interest? Do you know of other work in this area? **Come say hi!**

Credits: Kevin Lewi in particular, but also many conversations at HACS & RWC 2024, including Bas Westerbaan, Sophie Schmieg, Esha Ghosh, Alexander Scheel, Kevin Milner, Richard Barnes, and Brendan McMillion.

djackson@

moz://a