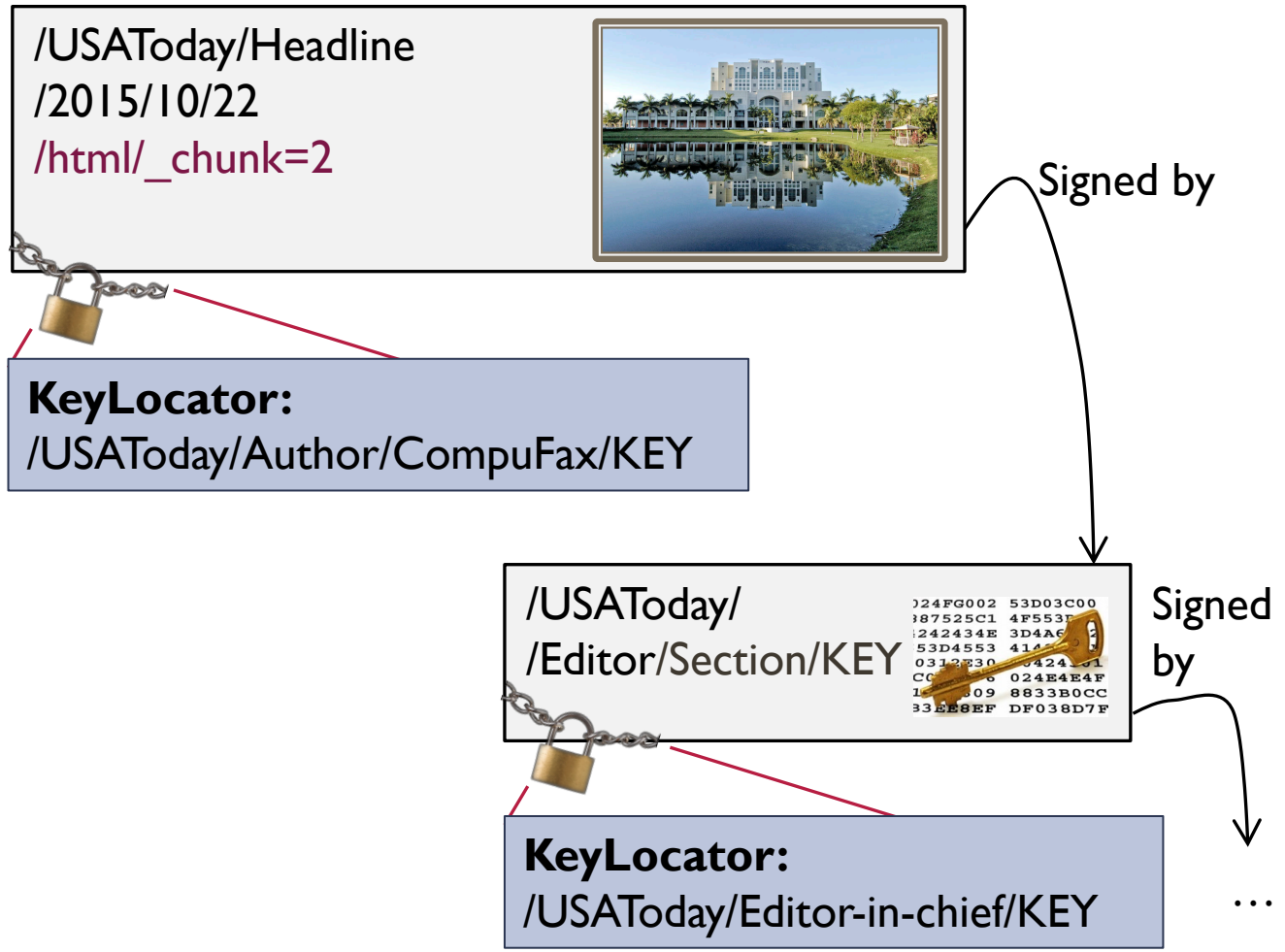
NDN DeLorean:
An Authentication System for Data Archives in Named Data Networking

Yingdi Yu (UCLA), **Alexander Afanasyev** (Florida International University),
Jan Seedorf (HFT Stuttgart), Zhiyi Zhang (UCLA), Lixia Zhang (UCLA),

ACM Information Centric Networking Conference,
September 27, 2017, Berlin, Germany

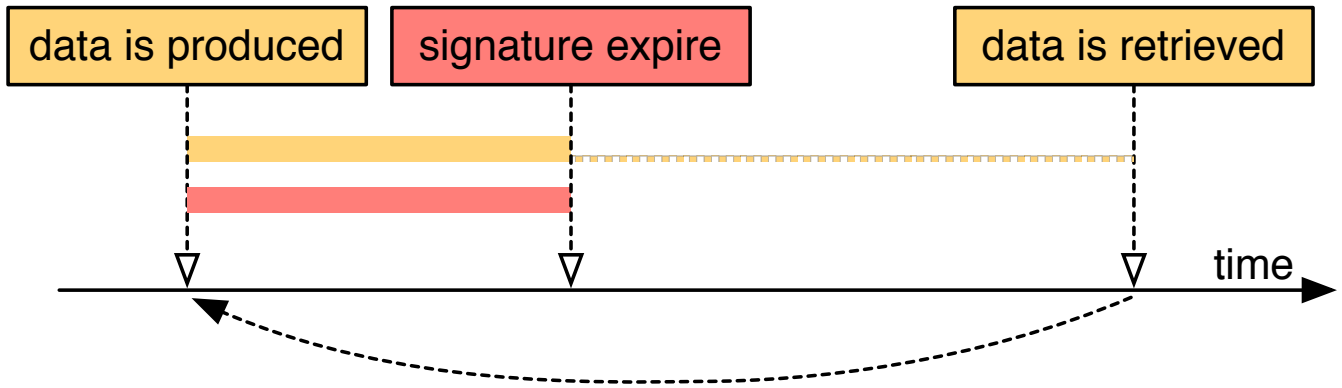
NDN and Data-Centric Security

- In NDN you sign the data with a digital signature..
  - ..so the users can check if they get the right data
- **Data secured both in motion and at rest**



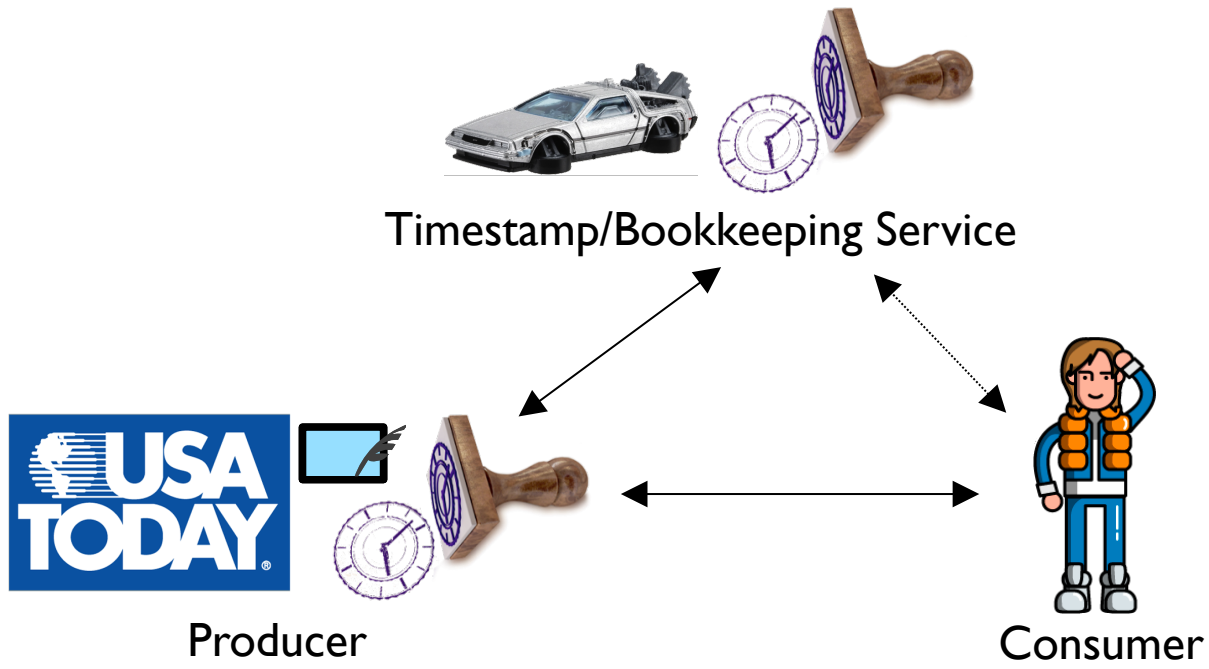
Mismatch Between Data and Signature Lifetimes

- **Data lifetime *can* be significantly longer than its signature's life time**
  - Parent certificate expiration or compromise, key compromise, crypto algorithm compromise, …
- Periodical re-signing unlikely to be feasible
  - Do not scale in long term
  - Data may outlive its producer
- Need a "look back" data authentication
  - Check signature validity at the time of data production



Look Back Data Authentication

- Need a certified timestamp for the past time point
  - ~~Trusted service~~
  - Hash-chain or block-chain
- DeLorean: Multi-level Merkle-tree based hash chain



DeLorean Workflow Overview

Publishers

Request proofs of signature existence

DeLorean Service

Retrieve proofs

Consumers

Retrieve data and part of volume chronicle as a proof; verify proof and signature

Aggregate requests and publish **"Chronicle Volumes"** for rolling time periods

Auditors

Audit published volumes

Security through publicity

Storage

2015-10-22 10:40am

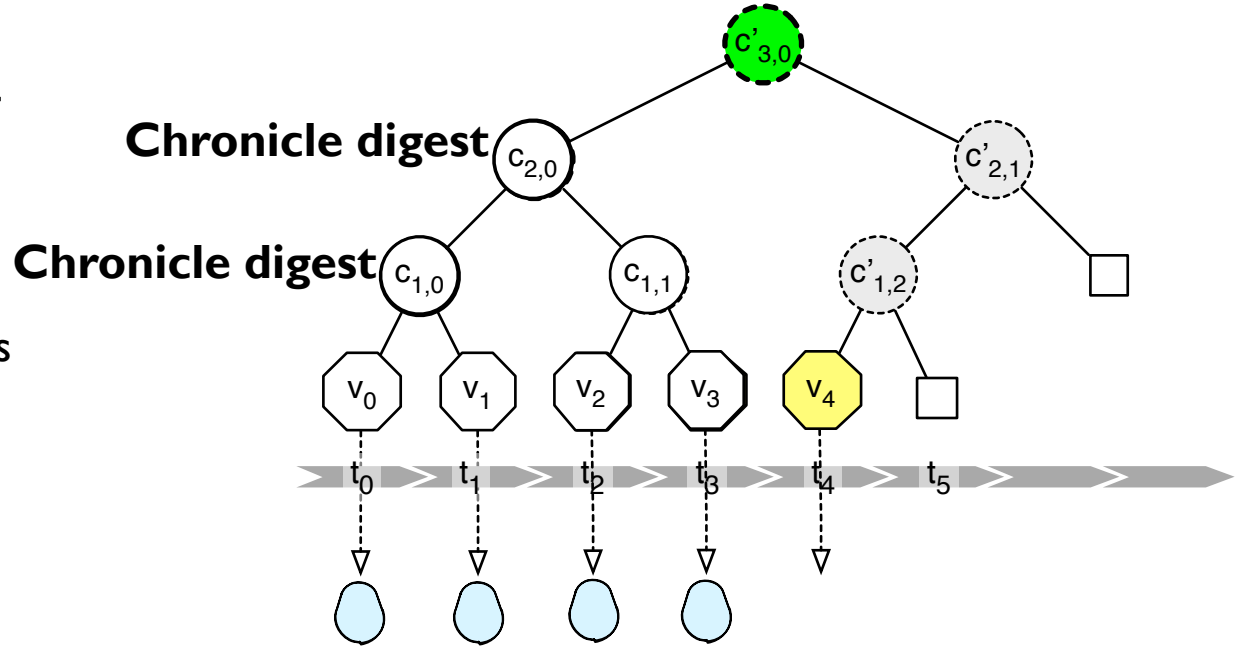
2015-10-22 10:50am

...

2016-05-10 3:30pm

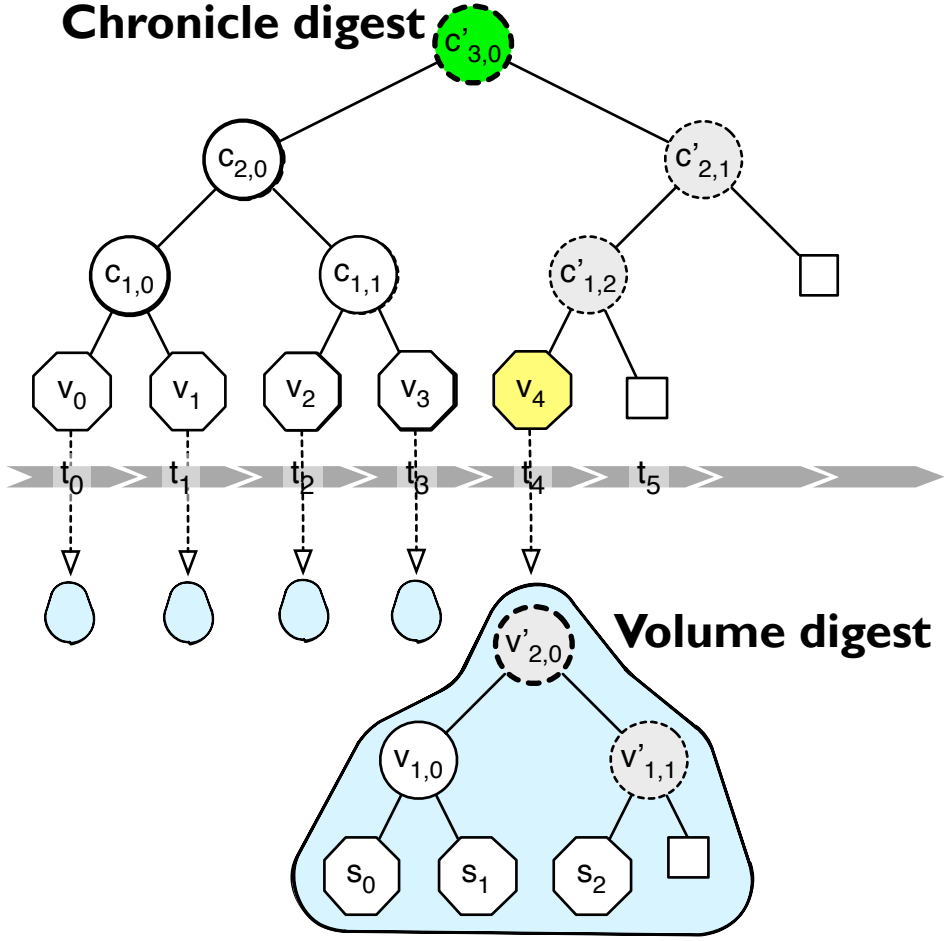
DeLorean Chronicle Tree Construction

- Chronicle
  - K-ary Merkle tree
  - Root hash (chonicle digest) fixes the state of the Chronicle
    - Each new volume updates nodes along the path to root
    - May create new root and intermediate nodes
- Existence verification:
  - $O(\log_k m)$
- Consistence verification:
  - $O(\log_k m)$
- Add/check volume to/in 20 year old 32-ary Chronicle with 10-min wide Volumes
  - 4 hash computations



DeLorean Chronicle Volume Construction

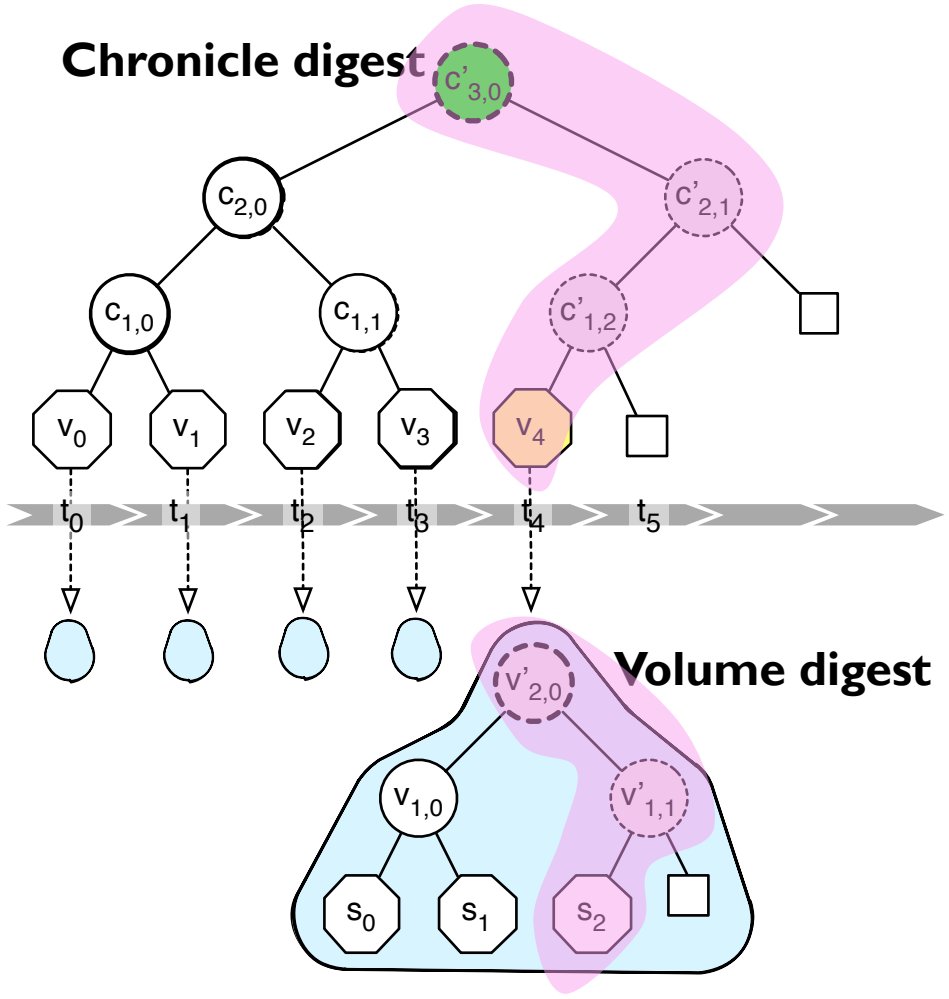
- Volume
  - K-ary Merkle tree
  - Root hash (volume digest) fixes the state of the time-specific volume
    - Each new signature updates nodes along the path to root
    - May create new root and intermediate nodes
- Existence verification:
  - $O(\log_k m)$
- Consistence verification:
  - $O(\log_k m)$
- Add/check signature to/in a Volume with > 1,000,000 signatures
  - 4 hash computations



Proof of Existence

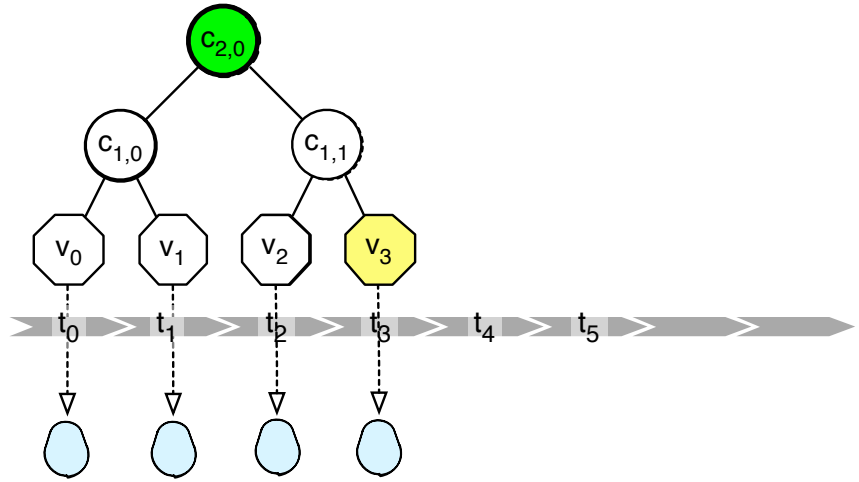
- Existence of volume v_4 in the Chronicle at time t_4
  - $\{ c'_{1,2}, c'_{2,1}, c'_{3,0} \}$
- Existence of signature s_2 in volume v_4 (at time t_4)
  - $\{ v'_{1,1}, c'_{2,0}, \}$

- Each node of chronicle and volume tree is published as NDN data packet
  - "Complete" nodes
    - Final version: all children are present and fixed
  - "Incomplete" nodes
    - Transient state
    - The latest transient node "fixes" all previous nodes

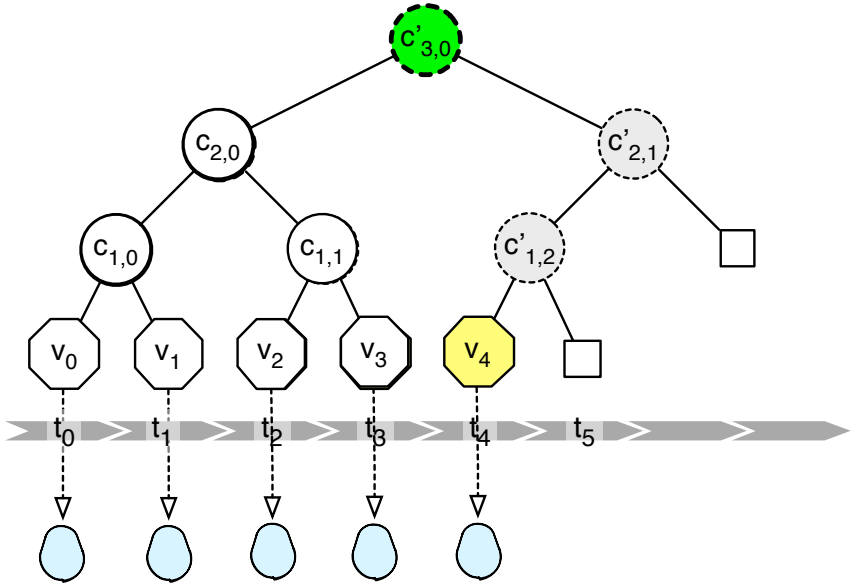


Proof of Consistency

- Periodic retrieval of current chronicle root
  - $c_{2,0}$, $c'_{3,0}$, $c''_{3,0}$, …
- Check if the "newer" root
  - incorporates the old root (if old root complete)
  - References the same subset of children (if incomplete)



- Easy to catch misbehavior
- Trivial networking and storage burden on auditors
  - Only root node needs to be stored
  - Usually one node retrieval



NDN Data Packet for DeLorean Nodes (Chronicle)

Signed by the DeLorean service for provenance only

- Naming convention
  - uniquely identify a node in a particular state of Chronicle and/or Volume tree

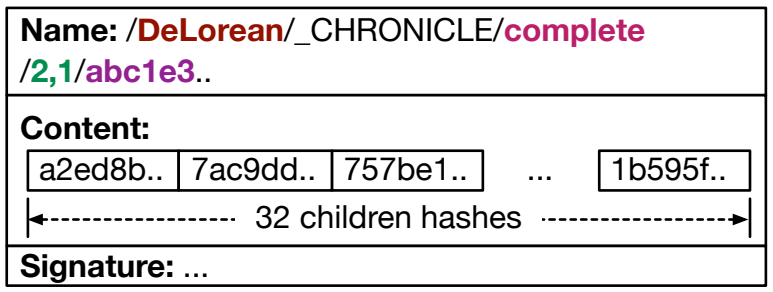
/[service-prefix]/_CHRONICLE/[NODE-STATE]/[layer],[index]/[hash]

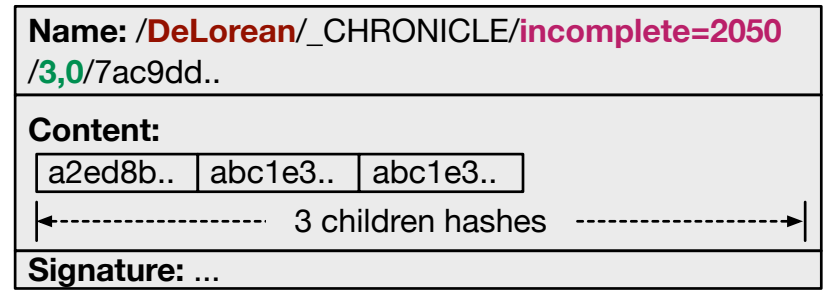
$$\text{“<NODE-STATE>”} = \begin{cases} \text{“complete”}, & \text{if } s \geq 32^l \times (i+1) \\ \text{“incomplete-(s+1)”}, & \text{otherwise} \end{cases}$$

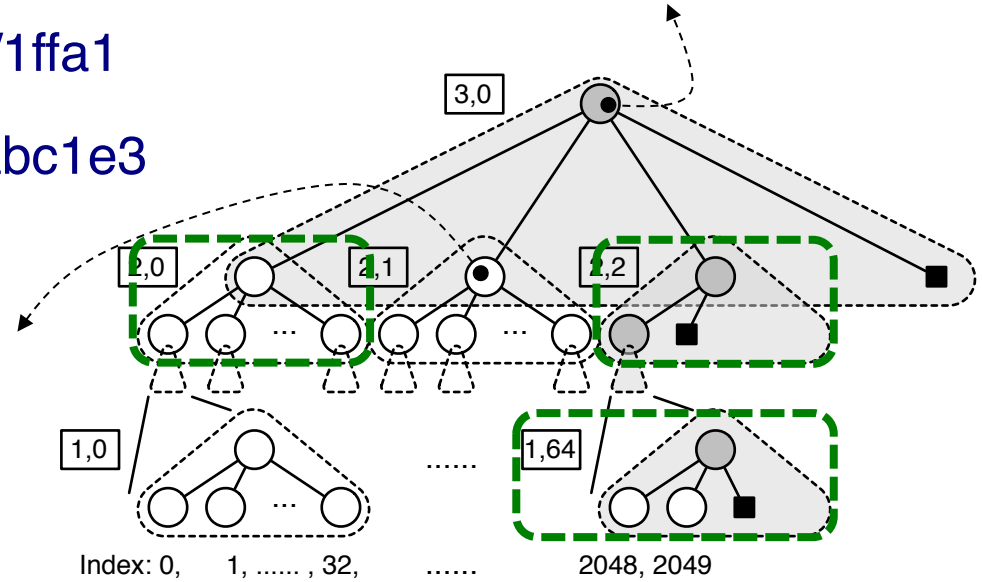
- Given a time point, the name of any node is determined

/DeLorean/_CHRONICLE/incomplete=2050/1,64/1ffa1

/DeLorean/_CHRONICLE/incomplete=2050/2,2/abc1e3
/DeLorean/_CHRONICLE/complete/2,0/a2ed8b







NDN Data Packet for DeLorean Nodes (Volume Trees)

Signed by the DeLorean service for provenance only

- Naming convention
  - uniquely identify a node in a particular state of Chronicle and/or Volume tree

/[service-prefix]/_VOLUME-[time-index]/[NODE-STATE]/[layer],[index]/[hash]

$$\text{“<NODE-STATE>”} = \begin{cases} \text{“complete”}, & \text{if } s \geq 32^l \times (i+1) \\ \text{“incomplete-(s+1)”}, & \text{otherwise} \end{cases}$$

Name: /DeLorean/_VOLUME-5/incomplete=2050 /3,0/7ac9dd..

Content:

a2ed8b..	abc1e3..	abc1e3..

3 children hashes

Signature: ...

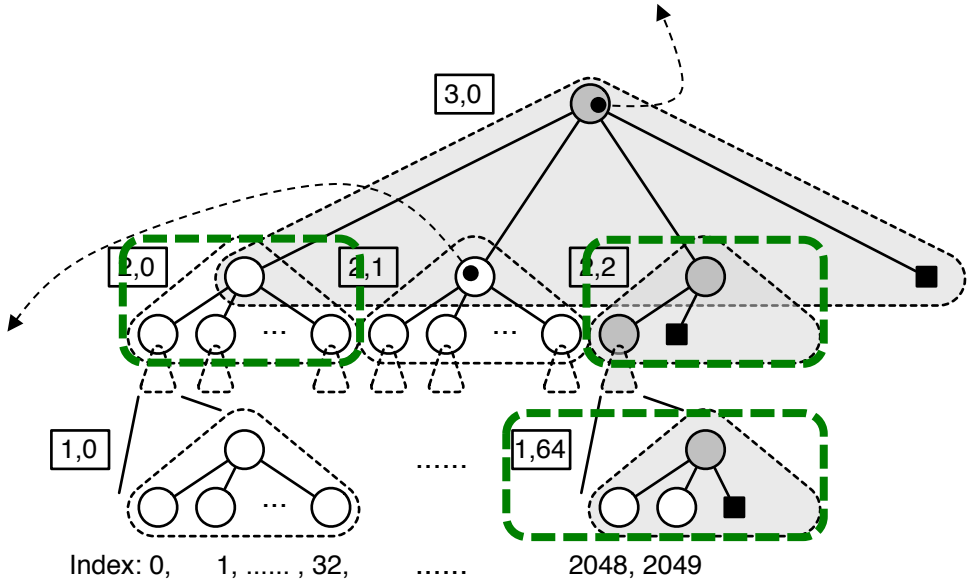
Name: /DeLorean/_VOLUME-5/complete /2,1/abc1e3..

Content:

a2ed8b..	7ac9dd..	757be1..	...	1b595f..

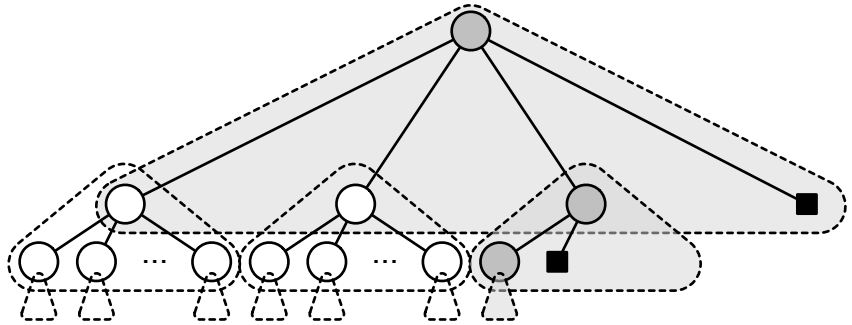
32 children hashes

Signature: ...

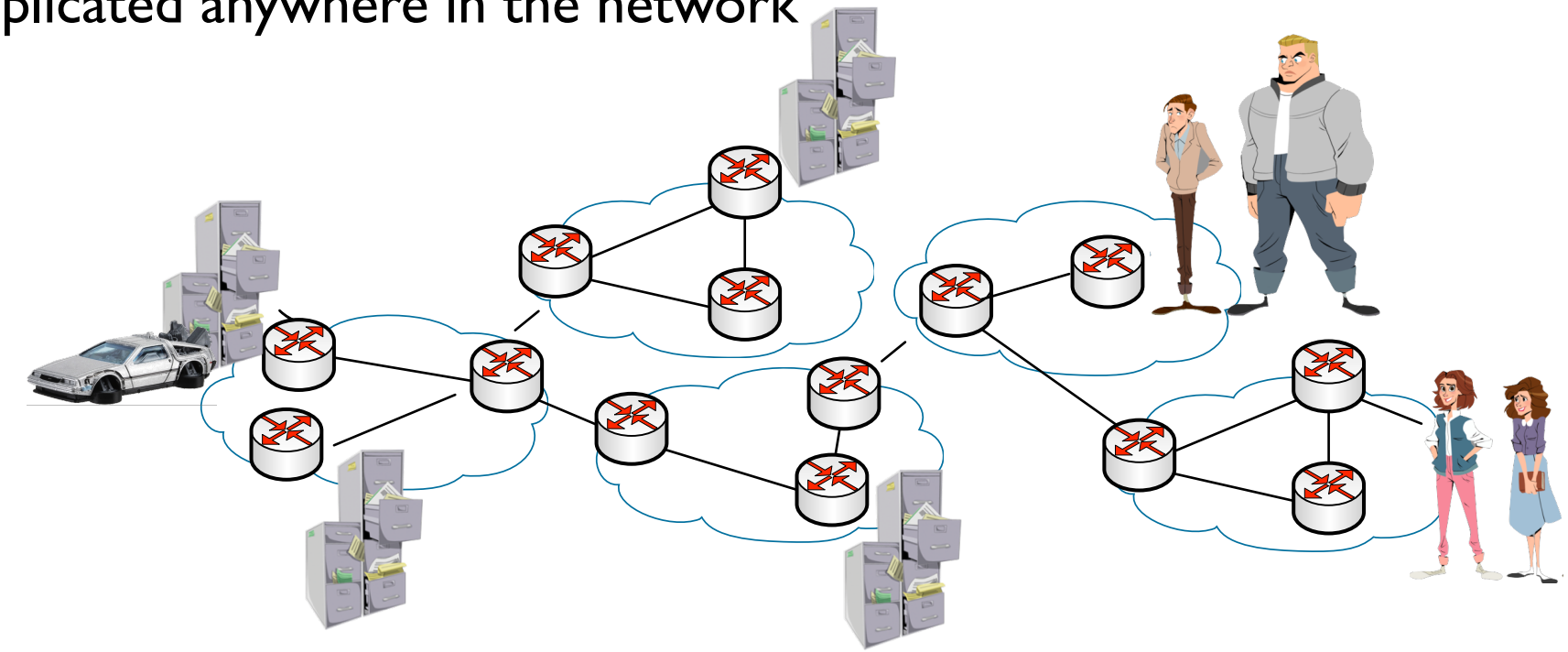


Node Retrieval

- Nodes at higher layers are frequently retrieved and benefit from caching
  - Complete never change
  - Most higher-level incomplete nodes don't change often

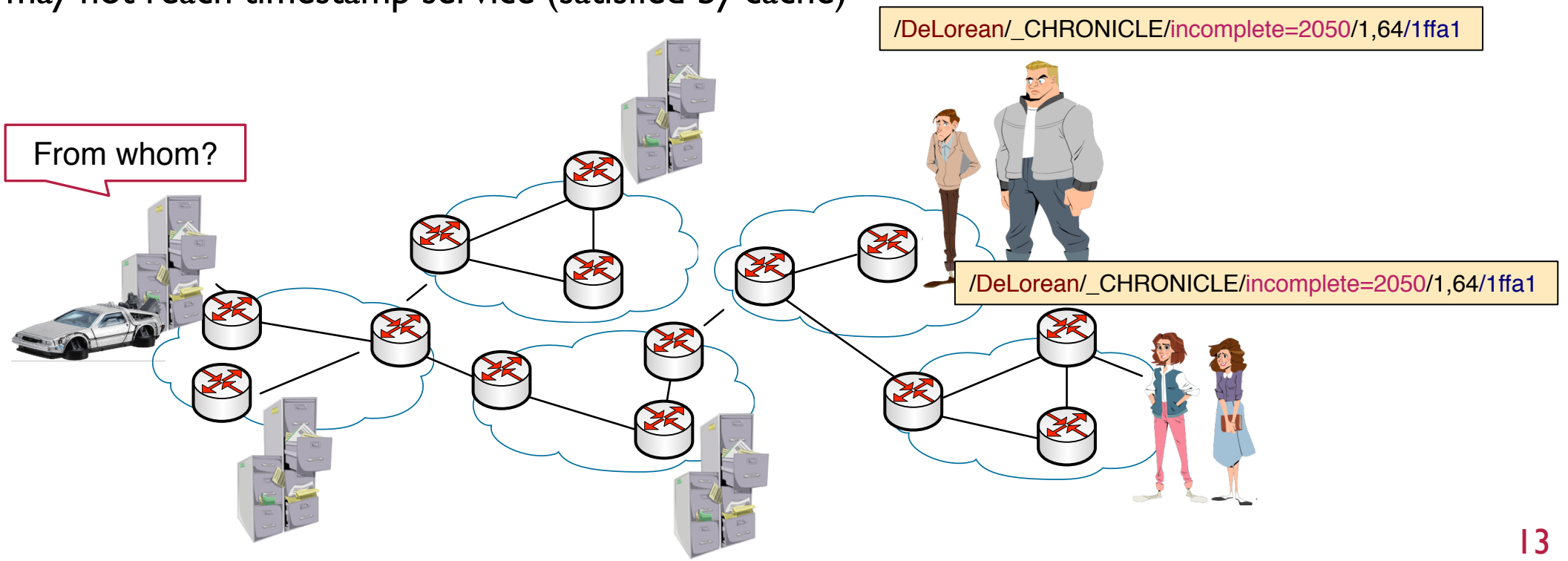


- Can be replicated anywhere in the network



Public Audit with Merkle Tree

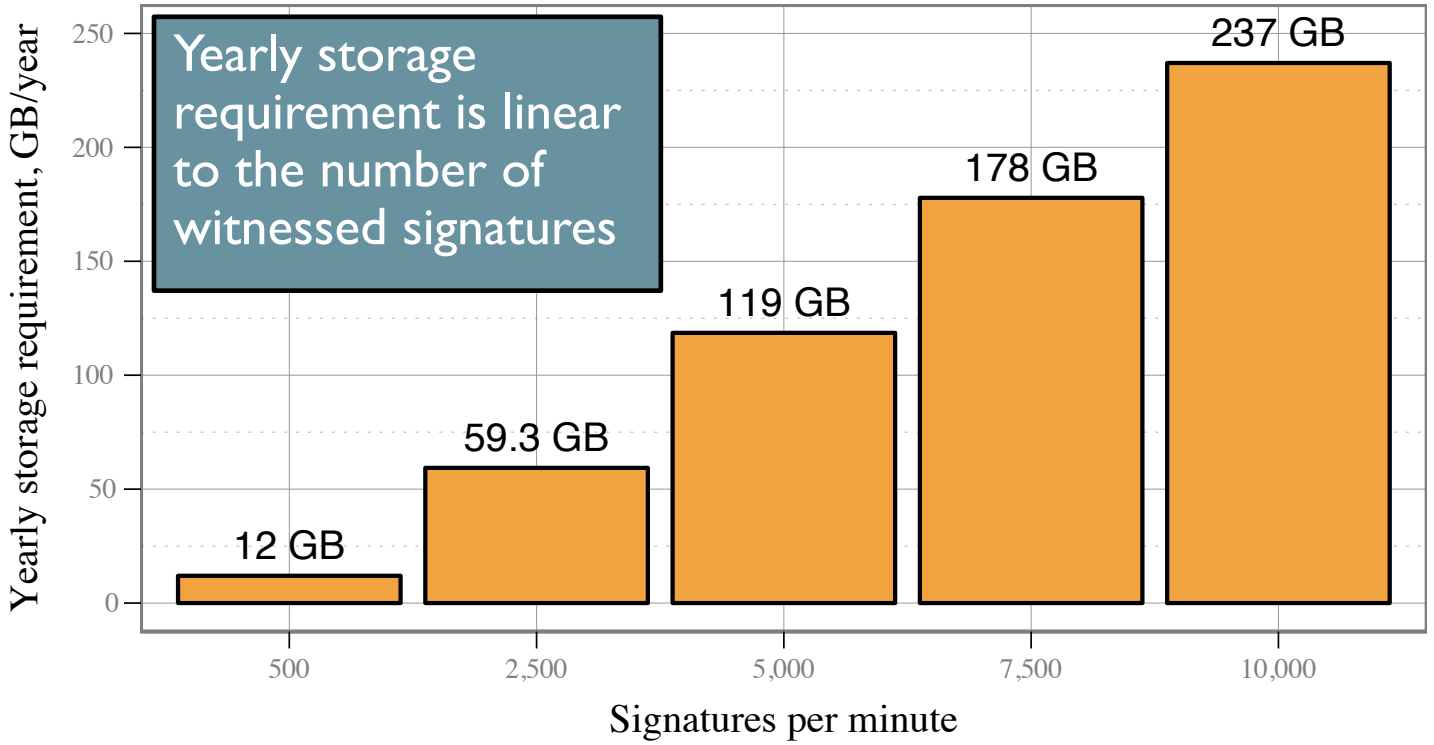
- All the users can verify consistence of the timestamp service
  - More users, the more secure and (publicly) reliable
  - Each published volume needs to be checked ~ at time of published to ensure timestamp trust
- Difficult to create double history
  - NDN interest does not carry sender address
  - Interest may not reach timestamp service (satisfied by cache)



Evaluation: Overview

- Analytical evaluations
  - Necessary storage capabilities at the DeLorean service provider
  - Verification cost at users
  - Needed number of auditors and audits per auditor
- Keep in mind …
  - Not every signature goes to DeLorean
  - For large volume archives, DeLorean need to track only signature of a manifest
- Real-world example: newspaper archive in public libraries
  - (based on data from statistica.com)
  - 700.000 pieces of newspaper content published per day
  - on average 486 pieces of content published per minute

Evaluation: DeLorean Service Storage Requirements



Yearly storage requirements at DeLorean service provider depending on signatures per minute
*(*Arity of Merkle tree: 32; duration of a timeslot within a volume: 10 minutes*)*

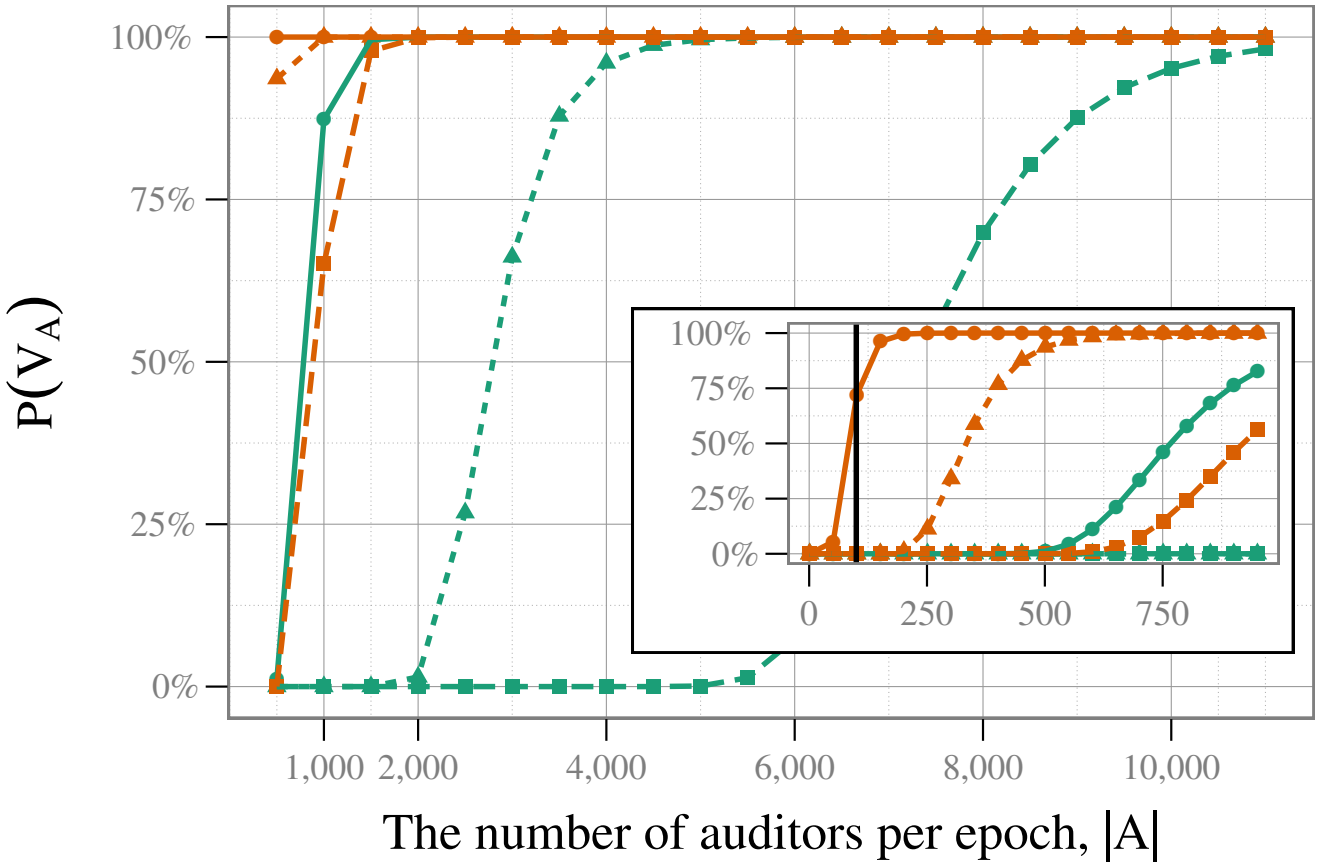
Amount of data a user would need to retrieve to verify the existence of a signature at a certain point in the past:

- For 32-ary trees and volume timeslot 10 minutes
  - 1500-byte data packet retrieval for the first 20 years
  - 4 x 1500-byte data packet retrieval for years 20-600

➢ Verification costs clearly negligible!

Evaluation: Required Number of Auditors

- Decentralized auditing
- Evaluation: probability that there is a volume that has not been verfiied by at least one auditor around the time the volume has been finalized



Epoch, days

1

3

7

Period during which each auditor fetches the chronlice at least once

Δ, minutes

10

60

Timeslot for volume creation

Discussion and Future Work

- Incentives to audit
  - Users
  - Providers
  - Competitors
- Recovery from inconsistency
  - "Bulletin boards" to post detected problems
    - Auditors cannot forge bad reports because of NDN signatures
  - Transition to new provider(s)
- Relation to real-time data production
  - Produce now, get proof of existence later

Summary

- With data-centric security, data lifetime can be longer than its signing key's validity period
- DeLorean provides publicly verifiable ("trust through transparency") bookkeeping service to enable "look back" validation of long-lived data
  - Collects signatures (signature digests) from producers
  - Publishes volumes of signatures collected within corresponding time periods
  - Efficient (storage, update, and lookup) Merkle-tree structure for signature volumes and chronicle of volumes

- Opportunities
  - Decouple the lifetime of data and signature
  - Make short-lived keys feasible