Sandia National Laboratories

Thien-Nam Dinh (thidinh@sandia.gov), Nicholas Pattengale (ndpatte@sandia.gov)

Composable Ledgers for Distributed Synchronic Web Archiving

Extended background paper ("The Synchronic Web") available online (https://arxiv.org/abs/2301.10733)

Motivation

Next-generation distributed archiving will require infrastructure to support the following capabilities:

- **Provenance** – metadata about the origins of data
- **Security** – cryptographic proofs for provenance

Existing solutions (e.g., blockchain) inadequately meet requirements for memory organizations:

- **Technical** – high scalability and interoperability
- **Economic** – low integration and maintenance cost

Approach

The Synchronic Web is a cryptographic capability affording secure provenance that is characteristically:

- **Performant** – scalable throughput and storage
- **Inexpensive** – freely available to all web clients

This work defines a Synchronic-Web-enabled procedure for securely sharing archive data using:

- **Split** – algorithm to commit and publish records
- **Merge** – algorithm to collect and verify records

Algorithm 1 Split Tree. Splits a verifiable map tree into a set of commitments. Algorithm 3 of the whitepaper defines *GetProof*

```
function SplitTree(map, tree)
    return [[key, value, GetProof(tree, key)]
        for each key, value ∈ map]
```

Algorithm 2 Merge Tree. Merges a set of commitments back into the original verifiable map tree

```
function MergeTree(entries)
    function Recurse(items, depth)
        if items.length = 1 ∧ items[0][2].length = depth then
            return [Hash(items[0][0] + items[0][1]), [∅, ∅]]
        zeros, ones ← [ ], [ ]
        for each item ∈ items do
            if Binary(item[0])[depth] = 0 then
                zeros.append(item)
            else
                ones.append(item)
        if zeros.length = 0 then
            left ← Recurse(zeros, depth + 1)
        else
            left ← [ones[0][2][depth], [∅, ∅]]
        if ones.length = 0 then
            right ← Recurse(ones, depth + 1)
        else
            right ← [zeros[0][2][depth], [∅, ∅]]
        return Node(
            Hash(left[0] + right[0]), [left, right])
    return Recurse(entries)
```

Progress

We implemented a **minimum viable prototype** to test the procedure across two component types:

- **Journals** – Synchronic Web client applications
- **Notaries** – Synchronic Web server infrastructure

We evaluated each component separately and achieved the following throughput at 1-second latency:

- **Client-side** – 100k tx/s from local data entries
- **Server-side** – 6k tx/s from remote clients requests

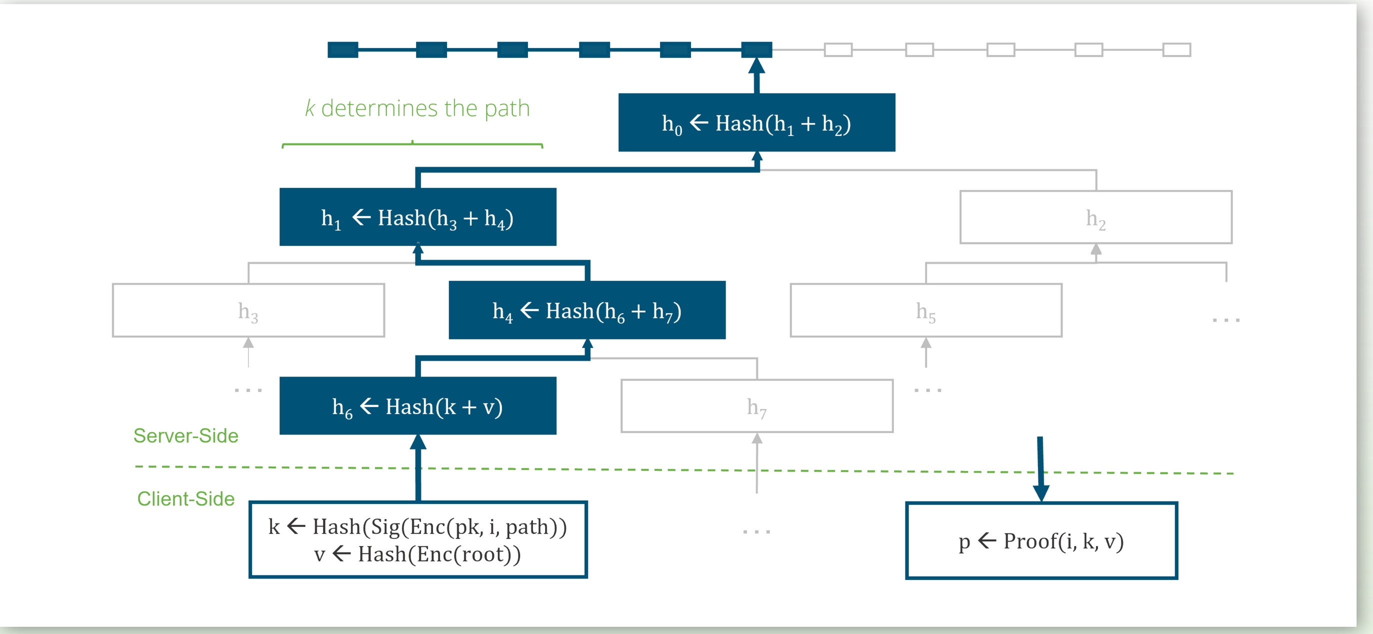
Next Steps

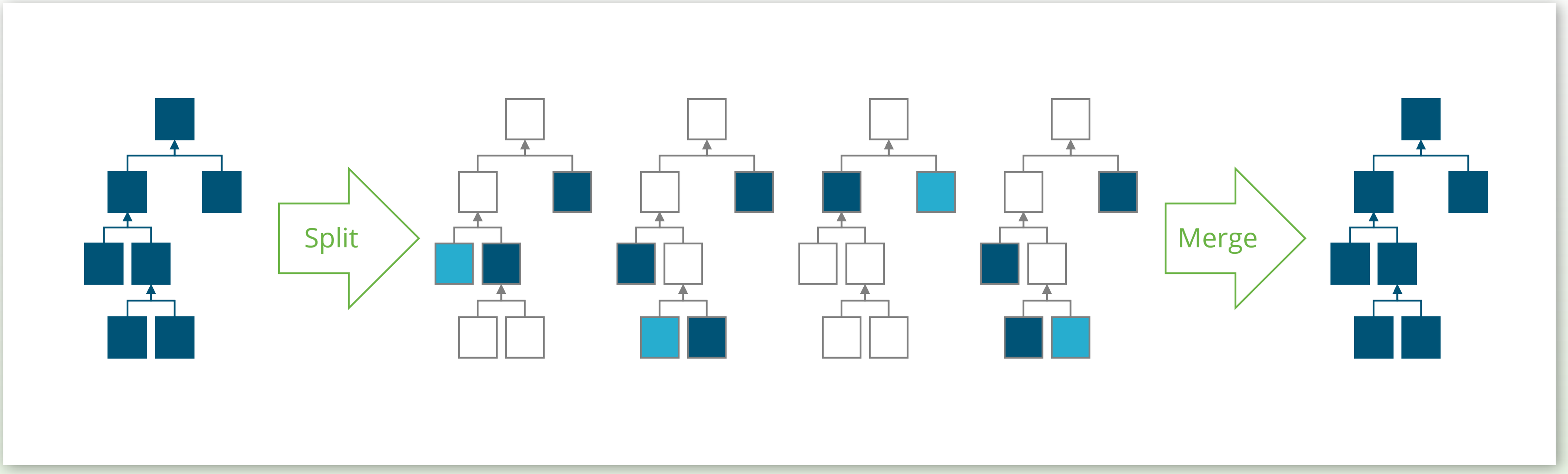
The future impact of this system can be improved through the following research directions:

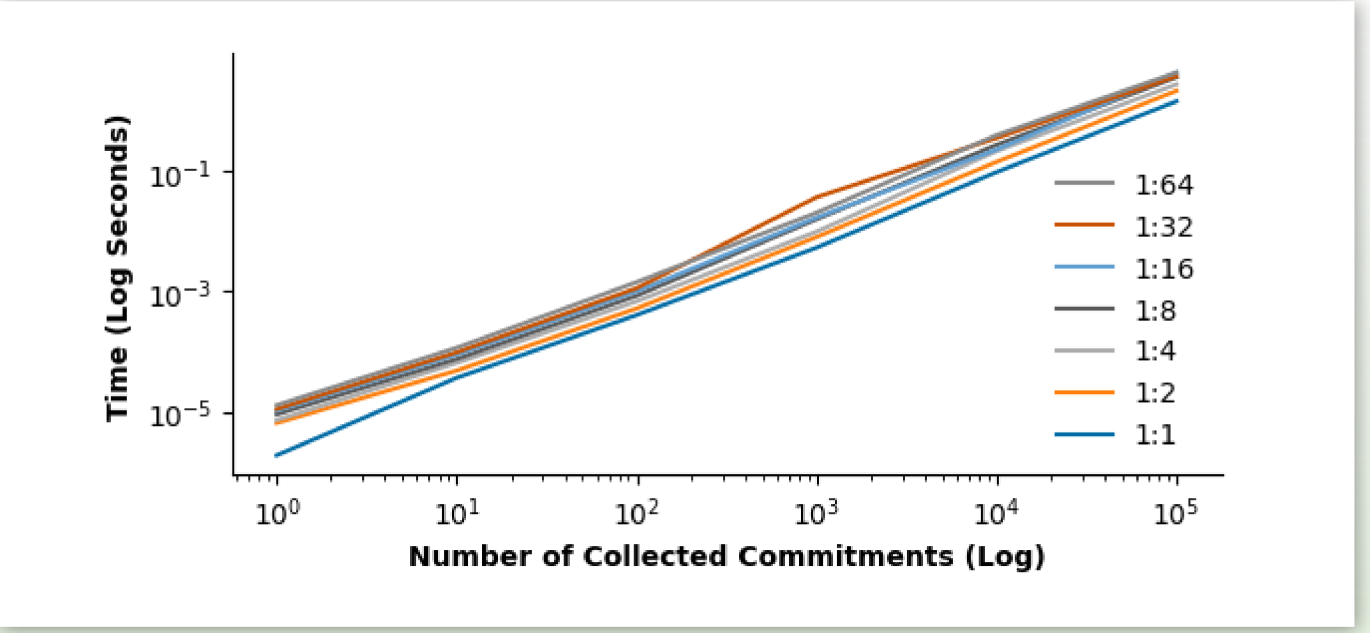
- **Cryptography** – more performant proof processing
- **Semantics** – more interoperable vocabularies

The team is pursuing steps to make Synchronic Web services **available as a free public good** through:

- **Software** – open-source code to build applications
- **Services** – public endpoints to secure the network







Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.