

# SoK: Cryptographic Authenticated Dictionaries

Harjasleen Malvai

**I** ILLINOIS / **IC3**

Francesca Falzon

**ETH** zürich

Andrew Zitek-Estrada\*

**EPFL**

Sarah Meiklejohn

**UCL**

Joseph Bonneau

**NYU**

# Background

# AuthDS Operations Part I

Like a dictionary but w/ proofs!

- 1) AuthDS.**Update** key / value pairs

Key	Value
Banana	Yellow
Peach	Orange

# AuthDS Operations Part I

Like a dictionary but w/ proofs!

- 1) AuthDS.**Update** key / value pairs
- 2) AuthDS.**Lookup** the value of a key + get proof

Main difference

Key	Value
Banana	Yellow
Peach	Orange

# AuthDS Operations Part I

Like a dictionary but w/ proofs!

- 1) AuthDS.**Update** key / value pairs
- 2) AuthDS.**Lookup** the value of a key + get proof (verify against commitment)

Main difference

- 3) **Commitment** (output by update)

State of the system

H

Key	Value
Banana	Yellow
Peach	Orange

Commitment is similar to a checksum!

# Applications

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

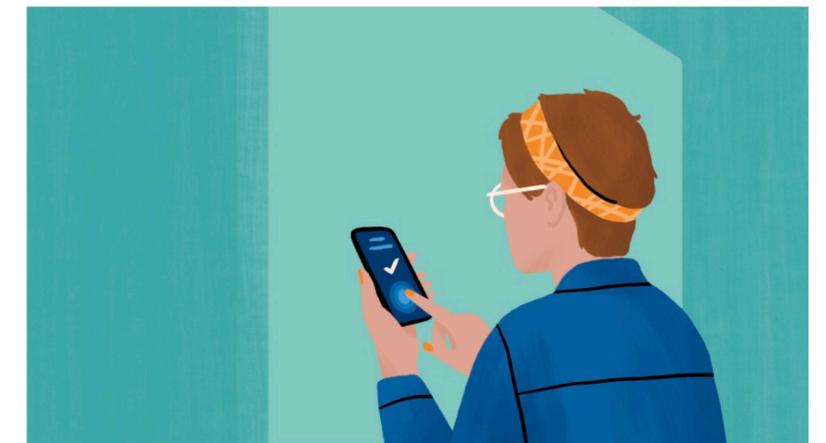
Key	Value
+1 (805) 978 8790	EncryptionKey

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

Key	Value
+1 (805) 978 8790	EncryptionKey

📦 Binary Transparency

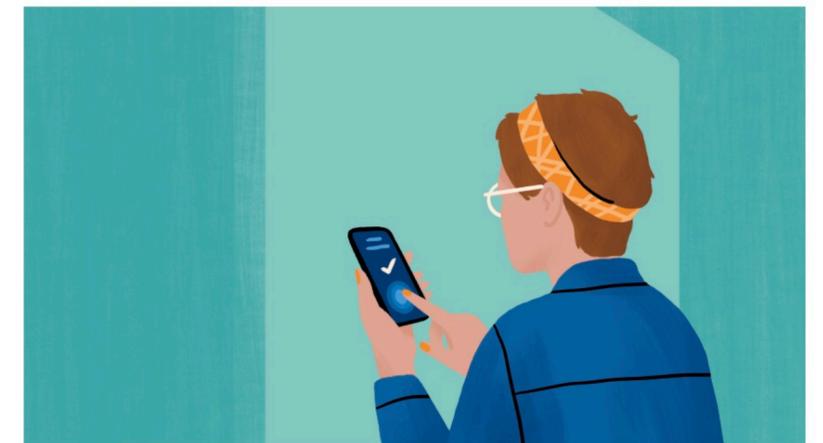
Release 4.1.13	SHA(Binary)
----------------	-------------

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

Key	Value
+1 (805) 978 8790	EncryptionKey

📦 Binary Transparency

Release 4.1.13	SHA(Binary)
----------------	-------------

🔑 Revocation Transparency

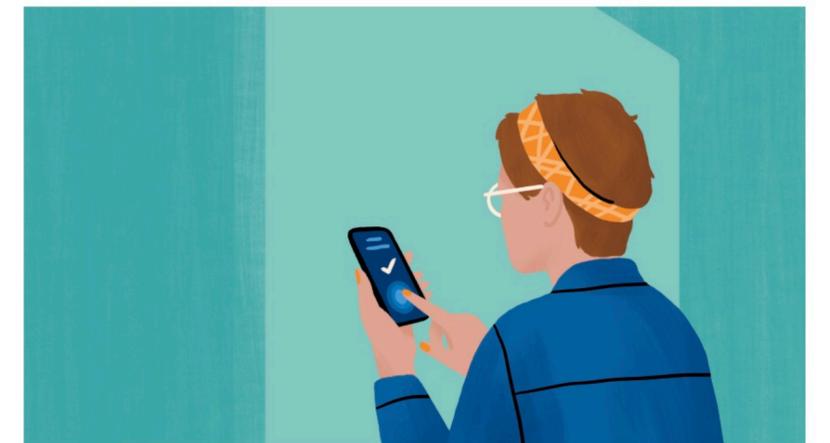
<a href="https://ndss.org">ndss.org</a>	PubCert
---	---------

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

Key	Value
+1 (805) 978 8790	EncryptionKey

📦 Binary Transparency

Release 4.1.13	SHA(Binary)
----------------	-------------

🔑 Revocation Transparency

<a href="https://ndss.org">ndss.org</a>	PubCert
---	---------

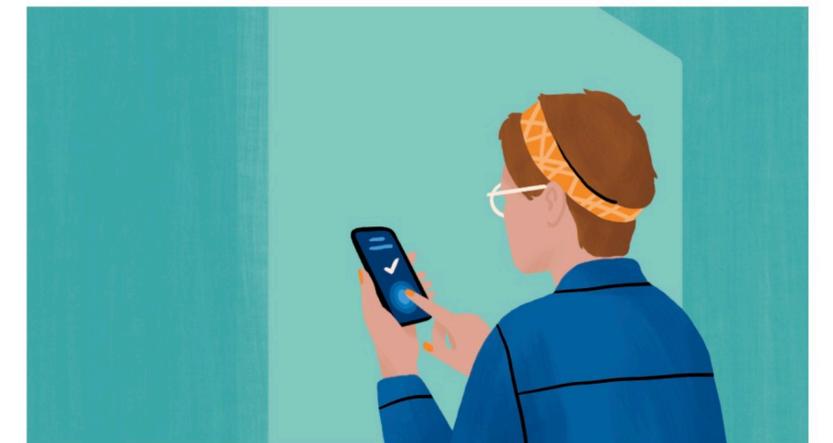
💾 Database and filesystem integrity

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

Key	Value
+1 (805) 978 8790	EncryptionKey

📦 Binary Transparency

Release 4.1.13	SHA(Binary)
----------------	-------------

🔒 Revocation Transparency

<a href="https://ndss.org">ndss.org</a>	PubCert
---	---------

💾 Database and filesystem integrity

💰 Blockchain

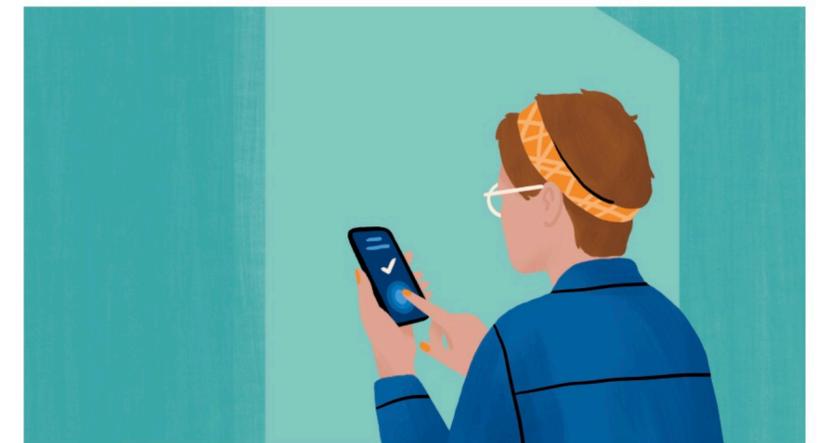
Identifier	Balance
------------	---------

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

# Global Consistency + Auditing

Is my view of the world the same as everyone else?



🗨️ Key Transparency

Key	Value
+1 (805) 978 8790	EncryptionKey

📦 Binary Transparency

Release 4.1.13	SHA(Binary)
----------------	-------------

🔑 Revocation Transparency

<a href="https://ndss.org">ndss.org</a>	PubCert
---	---------

💾 Database and filesystem integrity

💰 Blockchain

Identifier	Balance
------------	---------

📰 News Transparency

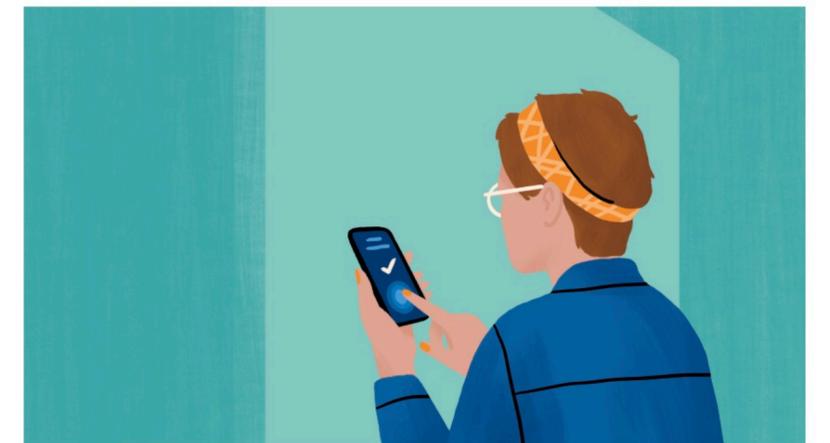
Article ID	SHA(Article)
------------	--------------

Engineering at Meta



POSTED ON APRIL 13, 2023 TO ANDROID, IOS, OPEN SOURCE, SECURITY & PRIVACY

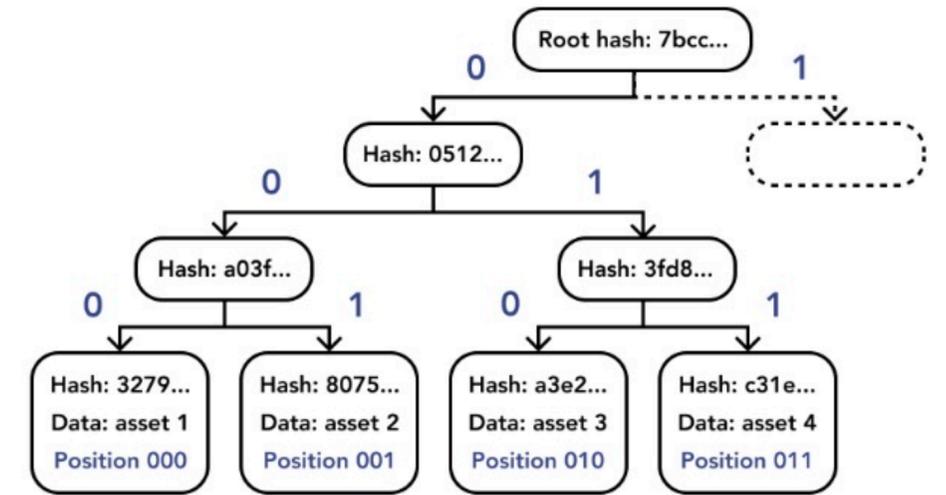
## Deploying key transparency at WhatsApp



By Sean Lawlor, Kevin Lewi

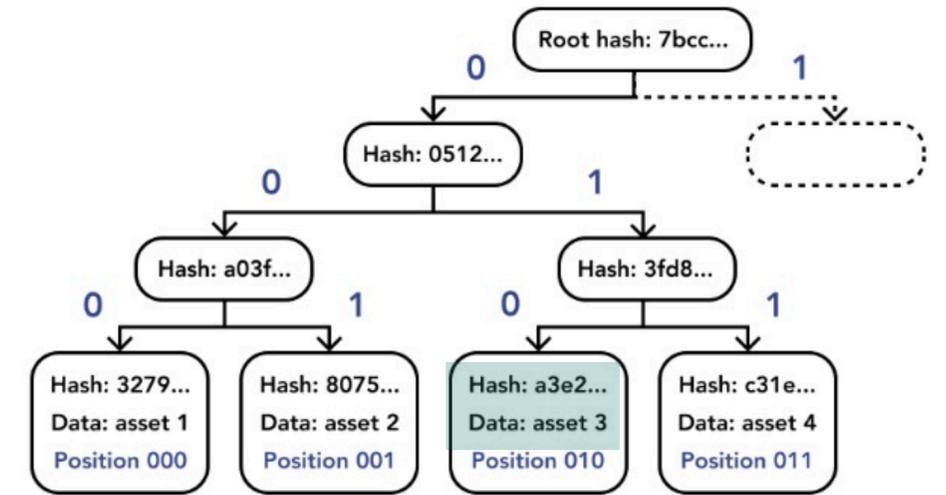
# AuthDS Operations Part II

## Must handle multiple versions



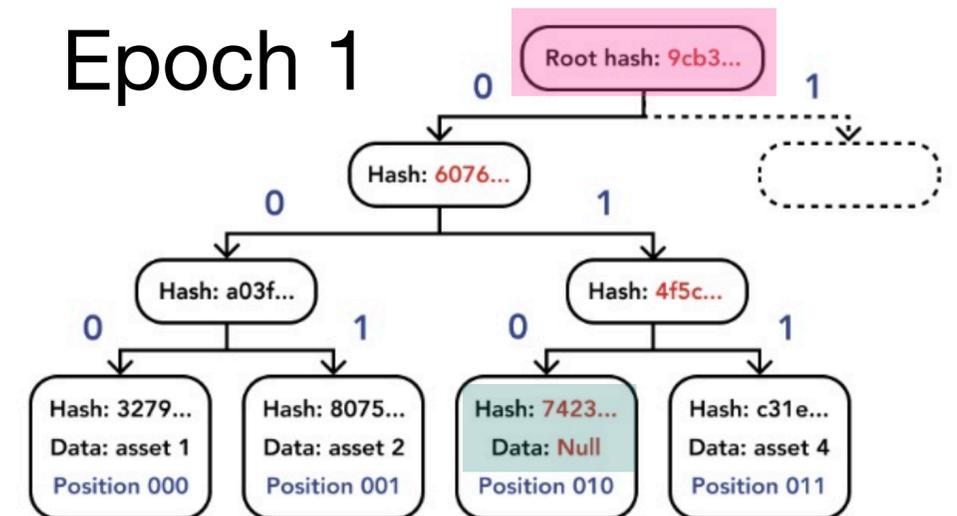
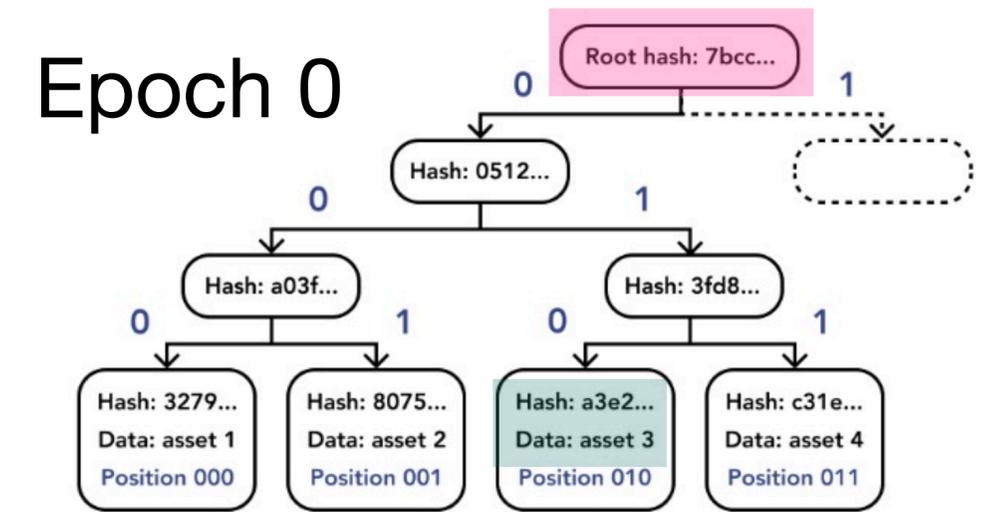
# AuthDS Operations Part II

## Must handle multiple versions



# AuthDS Operations Part II

## Must handle multiple versions



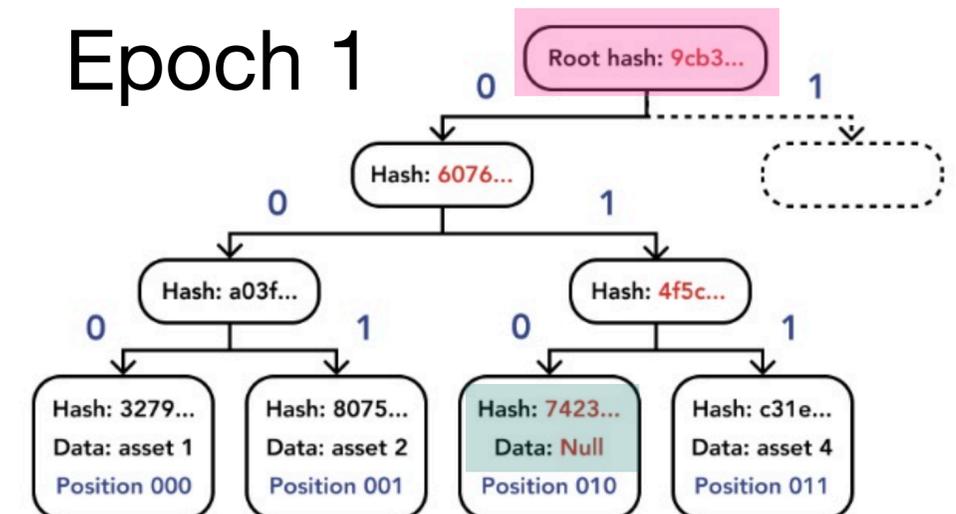
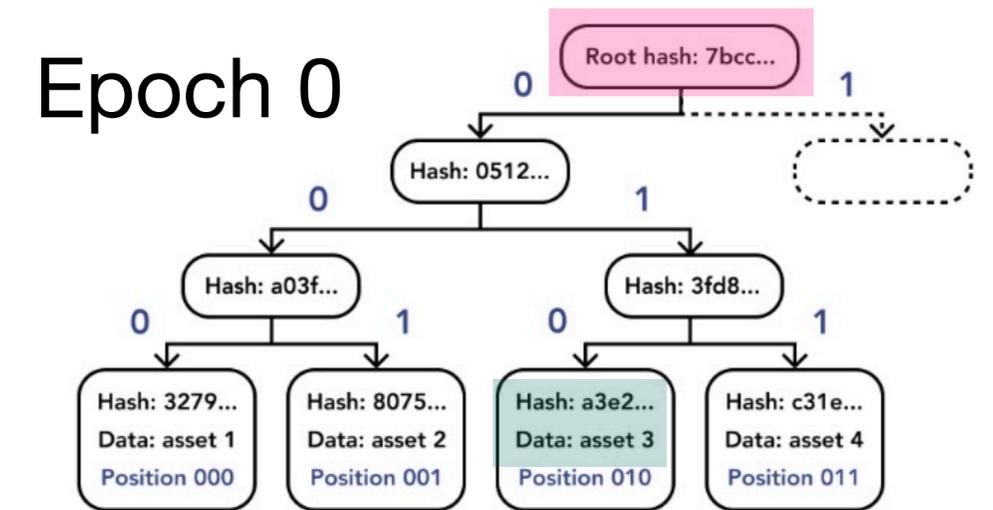
Handling mutations over time is technically challenging

# AuthDS Operations Part II

## Must handle multiple versions

- 4) **AuthDS.VerifyHistory** ← concerns one key

Value generated from a unique sequence of changes



Handling mutations over time is technically challenging

# AuthDS Operations Part II

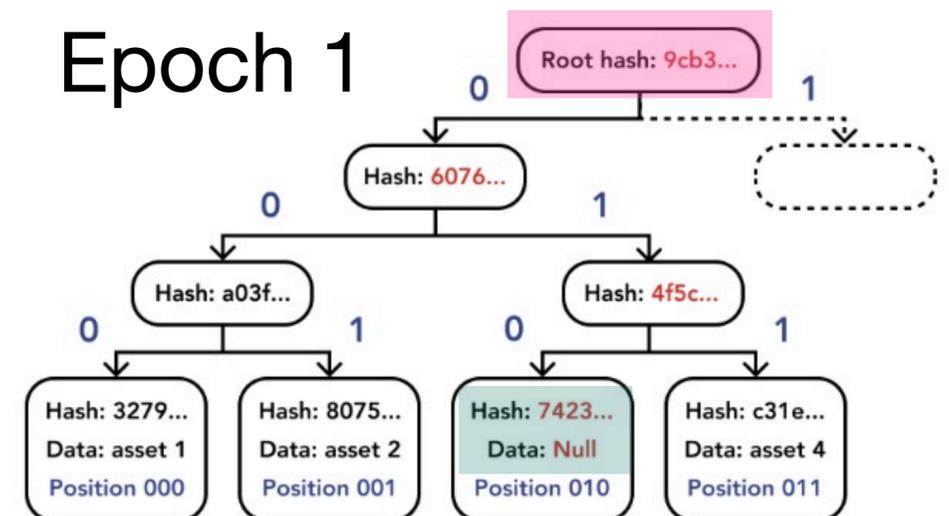
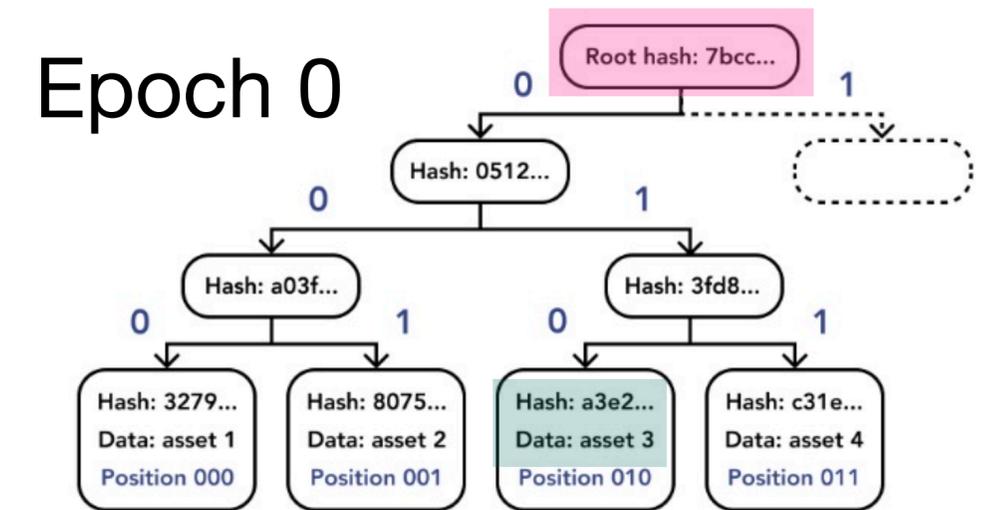
## Must handle multiple versions

- 4) **AuthDS.VerifyHistory** ← concerns one key

Value generated from a unique sequence of changes

- 5) **AuthDS.VerifyUpdate** ← concerns all keys

Current state is last state + sequence of all updates



Handling mutations over time is technically challenging

# Contributions

# Contributions at a Glance

## Four main contributions

(C1) Unified Framework for Trust Models

 Five models with five core roles each

# Contributions at a Glance

## Four main contributions

(C1) Unified Framework for Trust Models

 Five models with five core roles each

(C2) Map of Security Definitions

 Three core definitions and their relations

# Contributions at a Glance

## Four main contributions

(C1) Unified Framework for Trust Models

 Five models with five core roles each

(C2) Map of Security Definitions

 Three core definitions and their relations

(C3) Taxonomy of Constructions

 More than 30 schemes spanning 30 years

# Contributions at a Glance

## Four main contributions

(C1) Unified Framework for Trust Models

 Five models with five core roles each

(C2) Map of Security Definitions

 Three core definitions and their relations

(C3) Taxonomy of Constructions

 More than 30 schemes spanning 30 years

(C4) Asymptotic Performance Survey

 Tradeoff between lookup and update revealed

# Trust Models

# Five Trust Models

- 1) Private Outsourced Storage   
 Cloud backup, encrypted filesystems

## Private Outsourced Storage Model

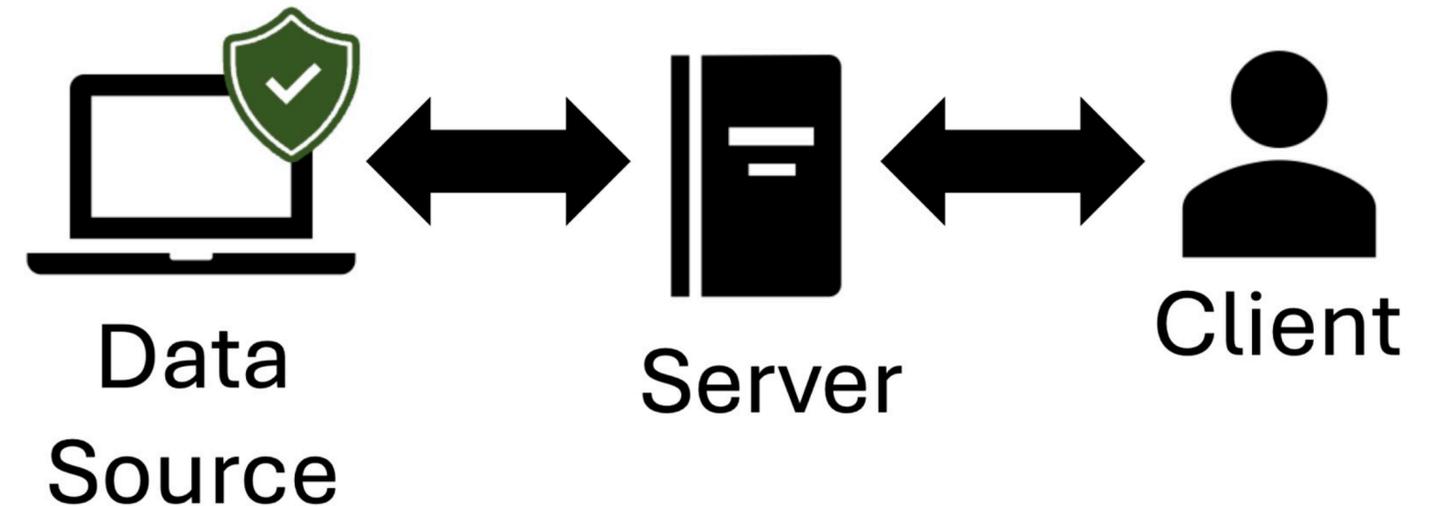


# Five Trust Models

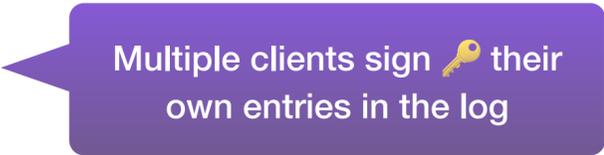
1) Private Outsourced Storage   
 Cloud backup, encrypted filesystems

2) Public Outsourced Storage   
 immudb, Amazon QLDB

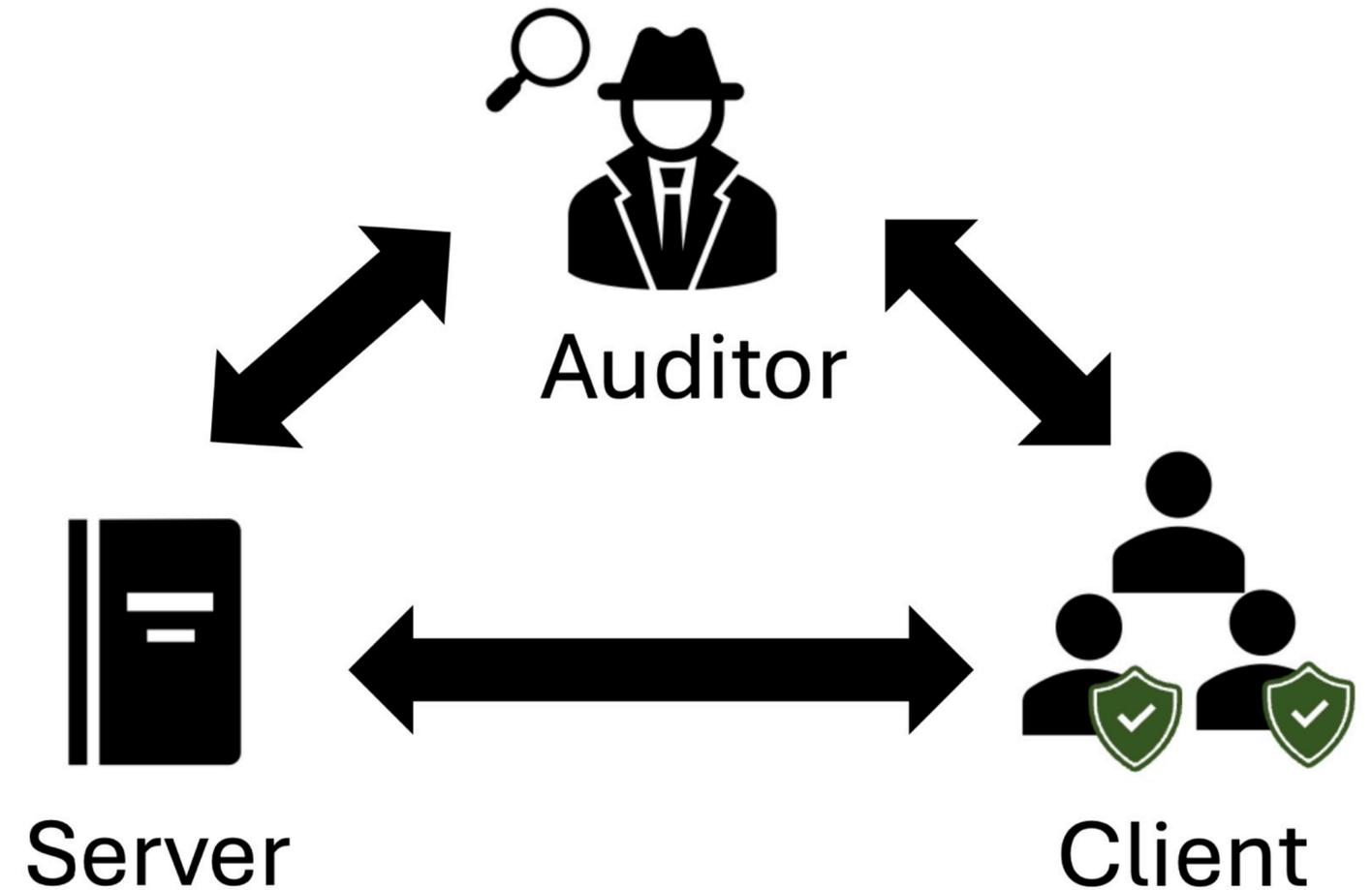
## Public Outsourced Storage Model



# Five Trust Models

- 1) Private Outsourced Storage   
☁️ Cloud backup, encrypted filesystems
- 2) Public Outsourced Storage   
 immudb, Amazon QLDB
- 3) Self-Sovereign  Keybase  
 Multiple clients sign  their own entries in the log

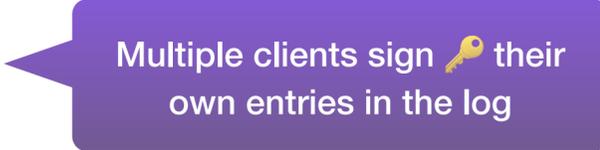
## Self-Sovereign Model

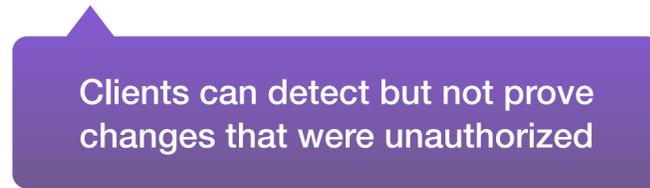


# Five Trust Models

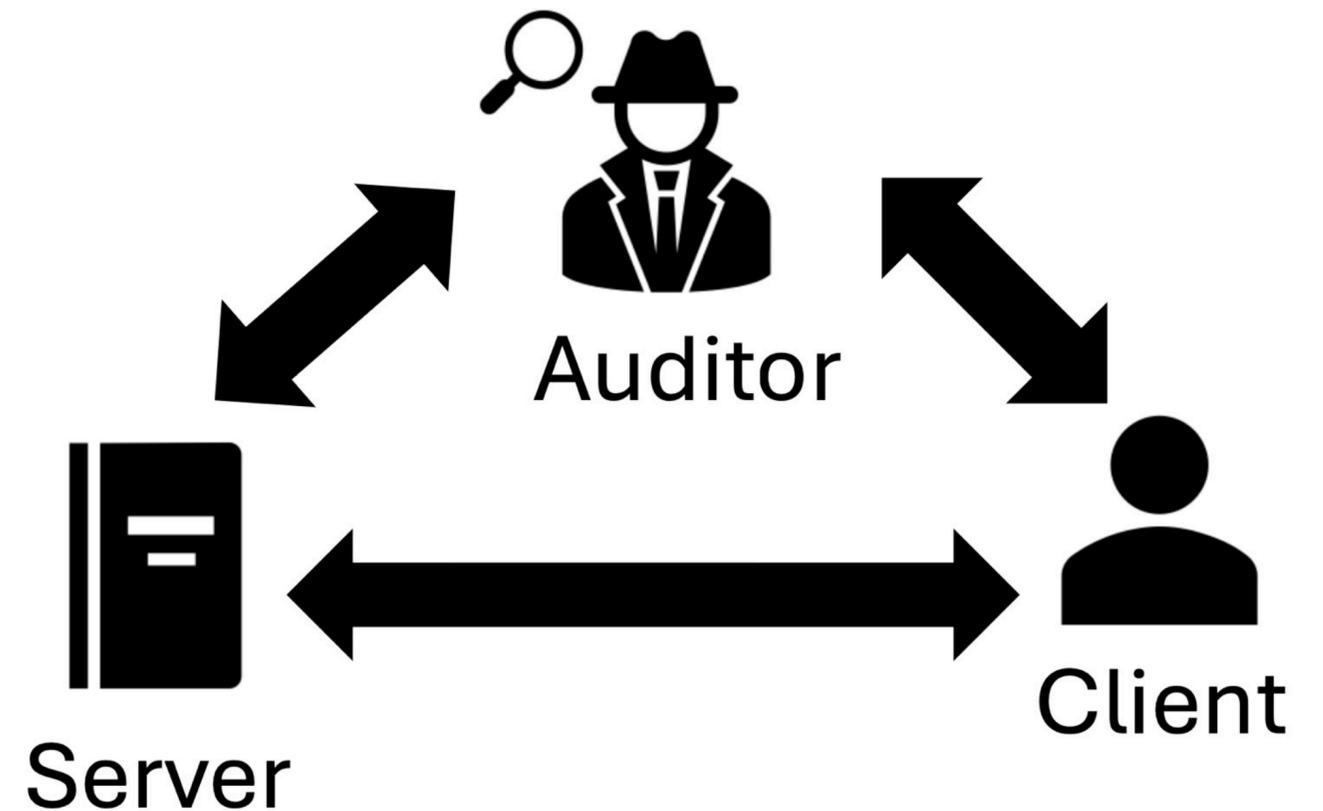
1) Private Outsourced Storage   
 Cloud backup, encrypted filesystems

2) Public Outsourced Storage   
 immudb, Amazon QLDB

3) Self-Sovereign  
 Keybase   
Multiple clients sign  their own entries in the log

4) Audited Transparency  
 WhatsApp KT, iMessage, Apple CT   
Clients can detect but not prove changes that were unauthorized

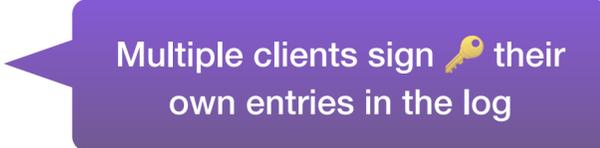
## Audited Transparency Model



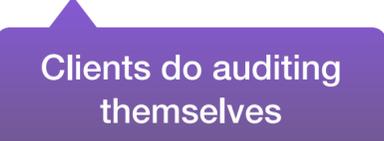
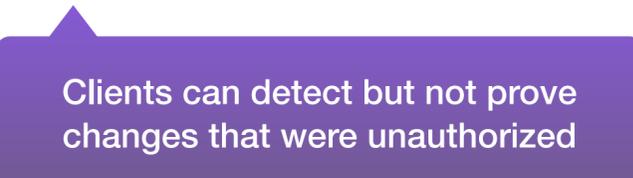
# Five Trust Models

1) Private Outsourced Storage   
 Cloud backup, encrypted filesystems

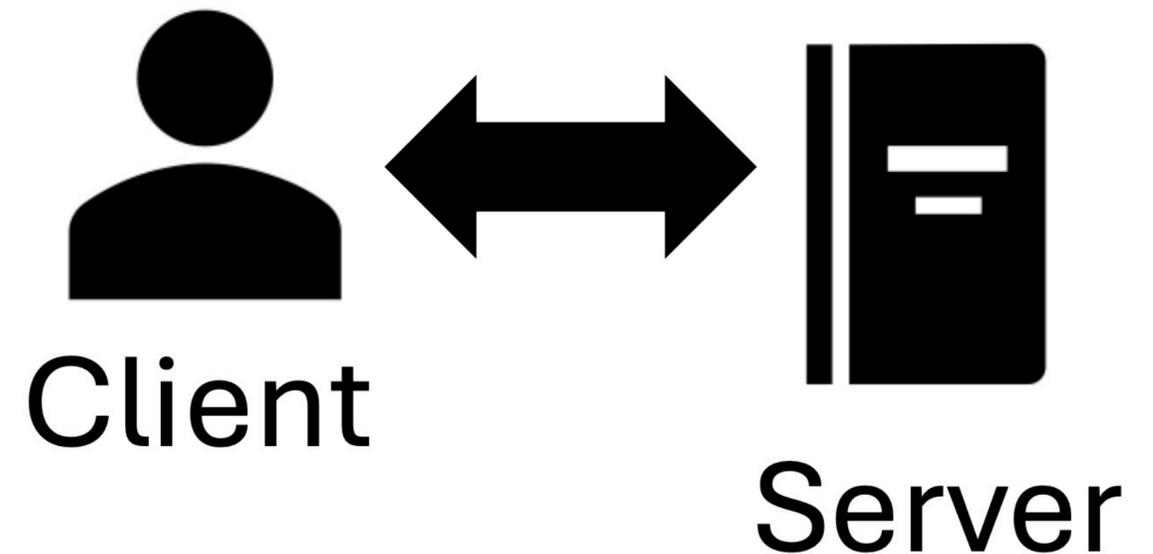
2) Public Outsourced Storage   
 immudb, Amazon QLDB

3) Self-Sovereign  
 Keybase   
Multiple clients sign  their own entries in the log

4) Audited Transparency  
 WhatsApp KT, iMessage, Apple CT

5) Transparency   
  
Clients do auditing themselves   
Clients can detect but not prove changes that were unauthorized

# Transparency Model



# Constructions

# Performance Barrier

Option A: Balanced

Lookup =  $O(\log n)$   
Update =  $O(\log n)$

Option B: Tradeoff

Lookup =  $O(1)$   
Update =  $O(n)$

No known construction achieves:  $O(\log n)$  lookup AND update

# Trust Does Not Buy Efficiency

Trusted Source

Lookup =  $O(\log n)$   
Update =  $O(\log n)$

No Trusted Party

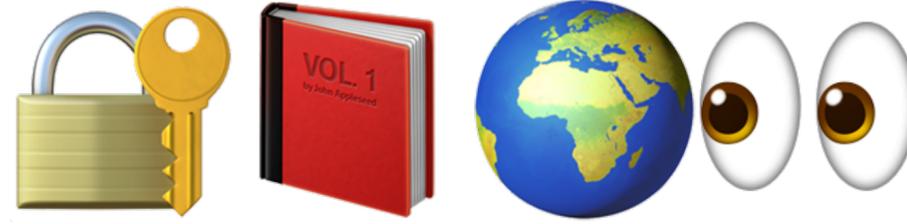
Lookup =  $O(\log n)$   
Update =  $O(\log n)$

Stronger trust  $\neq$  Better Asymptotics

# Discussion

# Open Questions

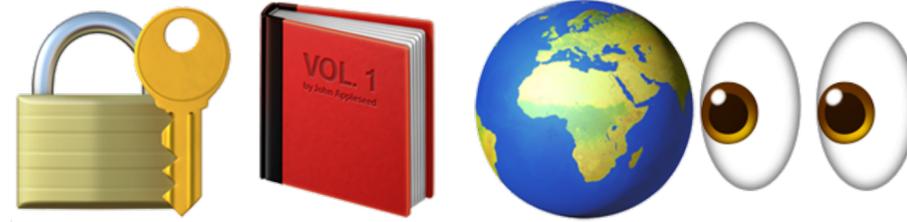
## And discussion



- How to incentivize auditing?

# Open Questions

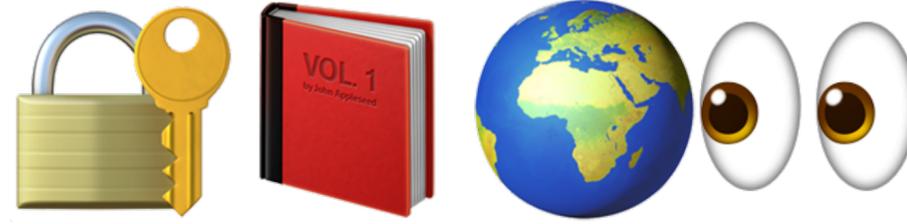
## And discussion



- How to incentivize auditing?
- Can we achieve sufficient performance for client auditing?

# Open Questions

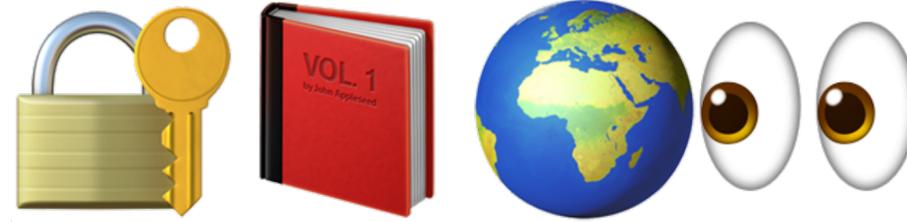
## And discussion



- How to incentivize auditing?
- Can we achieve sufficient performance for client auditing?
- Can we show reputation benefit from deploying ADs through user studies?

# Open Questions

## And discussion

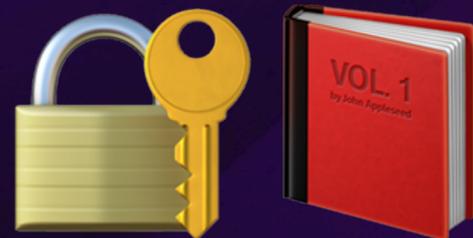


- How to incentivize auditing?
- Can we achieve sufficient performance for client auditing?
- Can we show reputation benefit from deploying ADs through user studies?
- How to handle regulatory compliance like Right to be Forgotten?

# Thank you!

See paper:

<https://shorturl.at/YQQTw>



# Three Types of ADs

Mutating Indexed ADs

CONIKS,  
Verkle, VeRSA

Append-only Indexed  
ADs

SEEMless,  
Parakeet, AAD

In-place append-only  
Indexed ADs

Keybase,  
Verdict, Chainiac

# Three Types of ADs

Mutating Indexed ADs

CONIKS,  
Verkle, VeRSA

Append-only Indexed  
ADs

SEEMless,  
Parakeet, AAD

In-place append-only  
Indexed ADs

Keybase,  
Verdict, Chainiac

Key difference: what is stored in each key?

# Three Types of ADs

Only last value

Mutating Indexed ADs

CONIKS,  
Verkle, VeRSA

Append-only Indexed  
ADs

SEEMless,  
Parakeet, AAD

In-place append-only  
Indexed ADs

Keybase,  
Verdict, Chainiac

Key difference: what is stored in each key?

# Three Types of ADs

Only last value

Mutating Indexed ADs

CONIKS,  
Verkle, VeRSA

Append-only Indexed  
ADs

SEEMless,  
Parakeet, AAD

In-place append-only  
Indexed ADs

Keybase,  
Verdict, Chainiac

All Values

Key difference: what is stored in each key?

# Three Types of ADs

Only last value

Mutating Indexed ADs

CONIKS,  
Verkle, VeRSA

Append-only Indexed  
ADs

SEEMless,  
Parakeet, AAD

All Values in  
separate structure

All Values

In-place append-only  
Indexed ADs

Keybase,  
Verdict, Chainiac

Key difference: what is stored in each key?

# Security Definitions

# Three Core Security Definitions

## 1) Value Binding (weakest)

- ✓ Two key lookups from same commitment get same answer

# Three Core Security Definitions

## 1) Value Binding (weakest)

- ✓ Two key lookups from same commitment get same answer
- ✗ Server changing key value between epochs

# Three Core Security Definitions

## 1) Value Binding (weakest)



- ✓ Two key lookups from same commitment get same answer
- ✗ Server changing key value between epochs

## 2) History Binding (stronger)

- ✓ Lookups are consistent with history verification

# Three Core Security Definitions

## 1) Value Binding (weakest)



- ✓ Two key lookups from same commitment get same answer
- ✗ Server changing key value between epochs

## 2) History Binding (stronger)

- ✓ Lookups are consistent with history verification
- ✗ Consistent, public, but *non client authorized* key changes

# Three Core Security Definitions

## 1) Value Binding (weakest)



- ✓ Two key lookups from same commitment get same answer
- ✗ Server changing key value between epochs

## 2) History Binding (stronger)



- ✓ Lookups are consistent with history verification
- ✗ Consistent, public, but *non client authorized* key changes

## 3) Read-Write Consistency (strongest)

- ✓ Lookups are consistent with history verification and changes were authorized

# Three Core Security Definitions

## 1) Value Binding (weakest)



- ✓ Two key lookups from same commitment get same answer
- ✗ Server changing key value between epochs

## 2) History Binding (stronger)

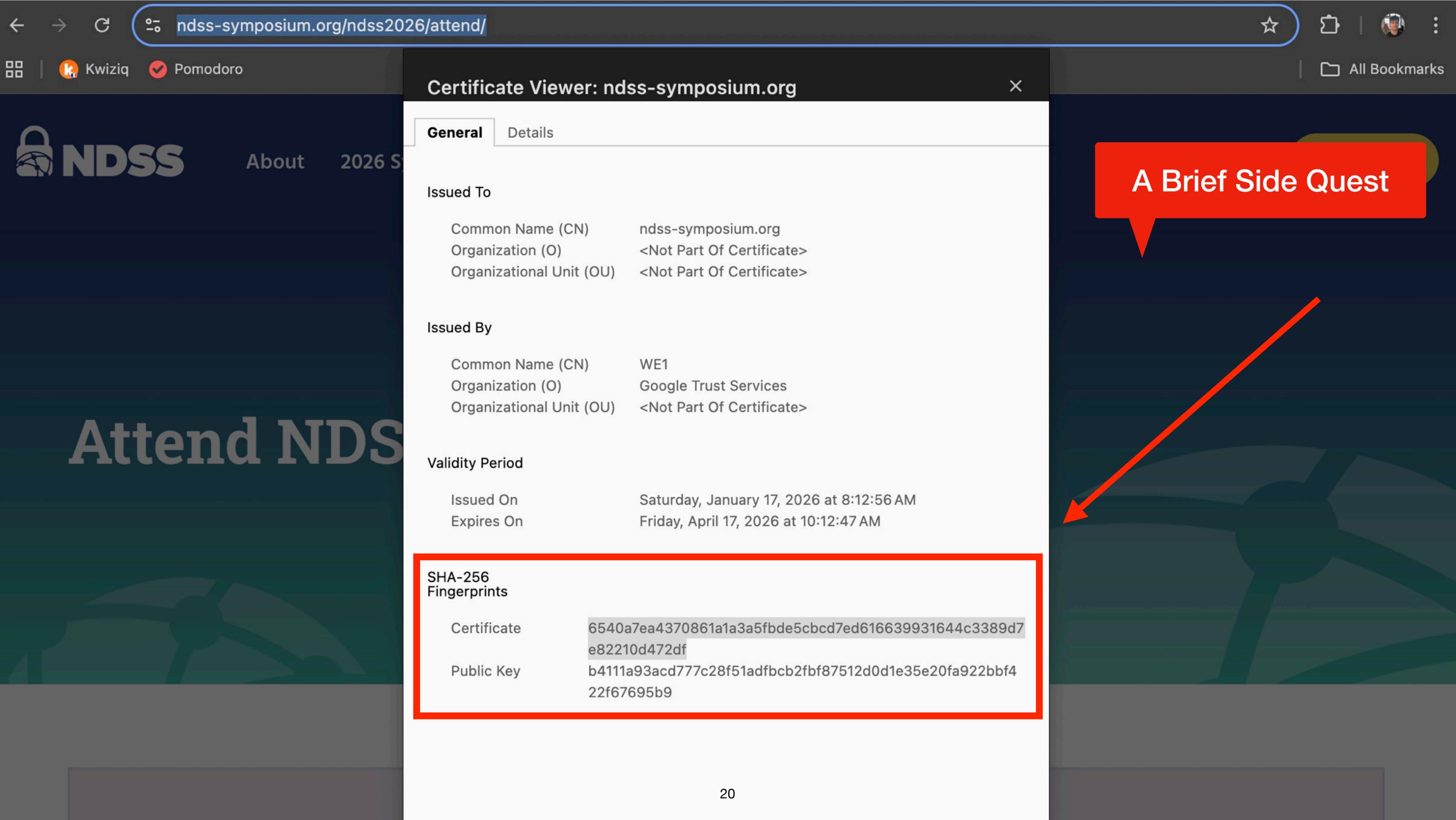


- ✓ Lookups are consistent with history verification
- ✗ Consistent, public, but *non client authorized* key changes

Best for No Trusted Source

## 3) Read-Write Consistency (strongest)

- ✓ Lookups are consistent with history verification and changes were authorized
- ✗ Doesn't apply to untrusted source models bc no authorization mechanism



### Certificate Viewer: ndss-symposium.org

General

Details

#### Issued To

Common Name (CN)	ndss-symposium.org
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

#### Issued By

Common Name (CN)	WE1
Organization (O)	Google Trust Services
Organizational Unit (OU)	<Not Part Of Certificate>

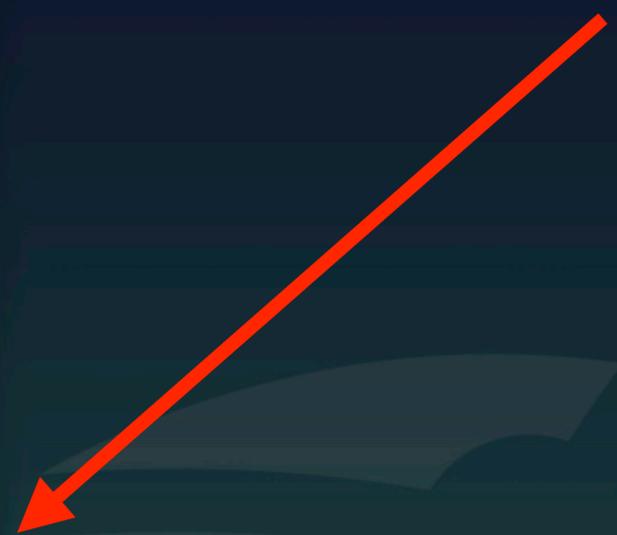
#### Validity Period

Issued On	Saturday, January 17, 2026 at 8:12:56 AM
Expires On	Friday, April 17, 2026 at 10:12:47 AM

#### SHA-256 Fingerprints

Certificate	6540a7ea4370861a1a3a5fbde5cbcd7ed616639931644c3389d7e82210d472df
Public Key	b4111a93acd777c28f51adfbc2fbf87512d0d1e35e20fa922bbf422f67695b9

A Brief Side Quest



Criteria SHA-256(Certificate) = '6540a7ea4370861a1a3a5fbde5cbcd7ed616639931644c3389d7e82210d472df'

**cert.sh ID** [23817853543](#)

**Summary** Leaf certificate

**Certificate Transparency**

Log entries for this certificate:

Timestamp	Entry #	Log Operator	Log URL
2026-01-17 08:15:31 UTC	1661073509	Google	https://ct.googleapis.com/logs/eu1/xenon2026h1
2026-01-17 08:15:31 UTC	1239888643	Sectigo	https://elephant2026h1.ct.sectigo.com
2026-01-17 08:15:31 UTC	885460556	IPng Networks	https://halloumi2026h1.mon.ct.ipng.ch
2026-01-17 08:15:31 UTC	1771769056	Google	https://ct.googleapis.com/logs/us1/argon2026h1

**A Brief Side Quest**

**Revocation**

[Report a problem](#) with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	<a href="#">Check</a>	?	n/a	?
CRL	The CA	Not Revoked	n/a	n/a	2026-02-09 13:09:10 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
<a href="#">OneCRL</a>	Mozilla	Not Revoked	n/a	n/a	n/a

**Certificate Fingerprints** **SHA-256** [6540A7EA4370861A1A3A5FBDE5CBCD7ED616639931644C3389D7E82210D472DF](#) **SHA-1** [A1F00AF6EC8F4EF92CB406D87D876082C67859BF](#)

**Certificate:**

Data:

Version: 3 (0x2)

**Serial Number:**  
cb:63:cb:10:9a:9b:ea:74:13:c8:80:04:1e:ca:c9:df

Signature Algorithm: ecdsa-with-SHA256

**Issuer:** (CA ID: 286236)

commonName = WE1  
organizationName = Google Trust Services  
countryName = US

Validity

Not Before: Jan 17 07:12:56 2026 GMT  
Not After : Apr 17 08:12:47 2026 GMT

Subject:

commonName = ndss-symposium.org

**Subject Public Key Info:**

Public Key Algorithm: id-ecPublicKey  
Public-Key: (256 bit)

**11 logs: Sectigo, TrustAsia, Cloudflare...**