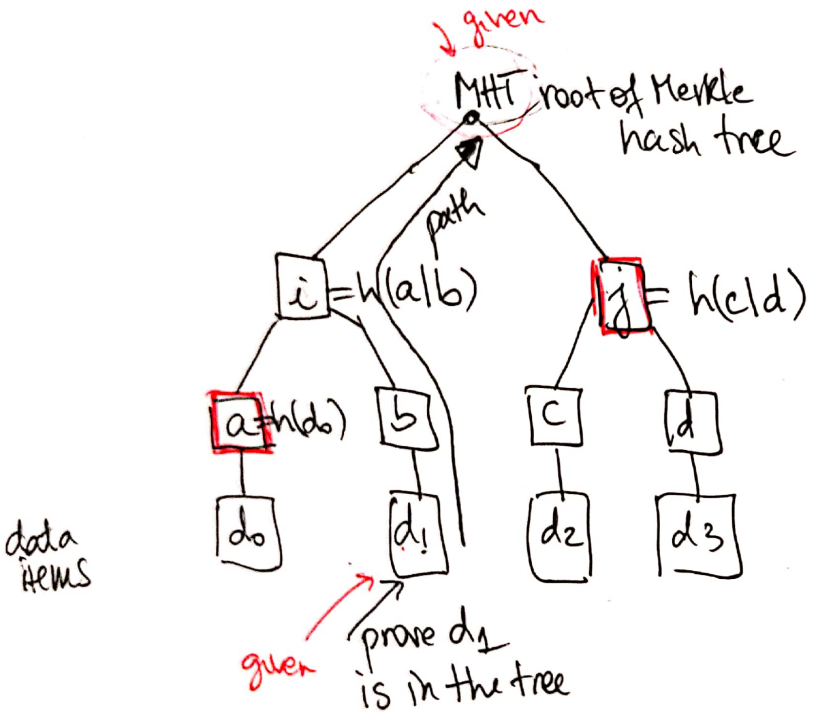
Merkle Trees (1979)

- hash is collision-resistant



$d_1 \xrightarrow{h} b \Big\} \xrightarrow{h} i \Big\} \xrightarrow{h}$ MHT ?= MHT you had

a, j

$h(h(a, h(d_1)), j) \stackrel{?}{=}$ MHT

Property: Given MHT, one can prove that data item d_1 is in the tree in a logarithmic # of steps and using a log number of nodes in the tree

Audit proof for a node N = siblings of nodes on the path from N to root

Client: have MHT

Attacker: says $d_1' \neq d_1$ is in the tree, a', j' is the audit proof

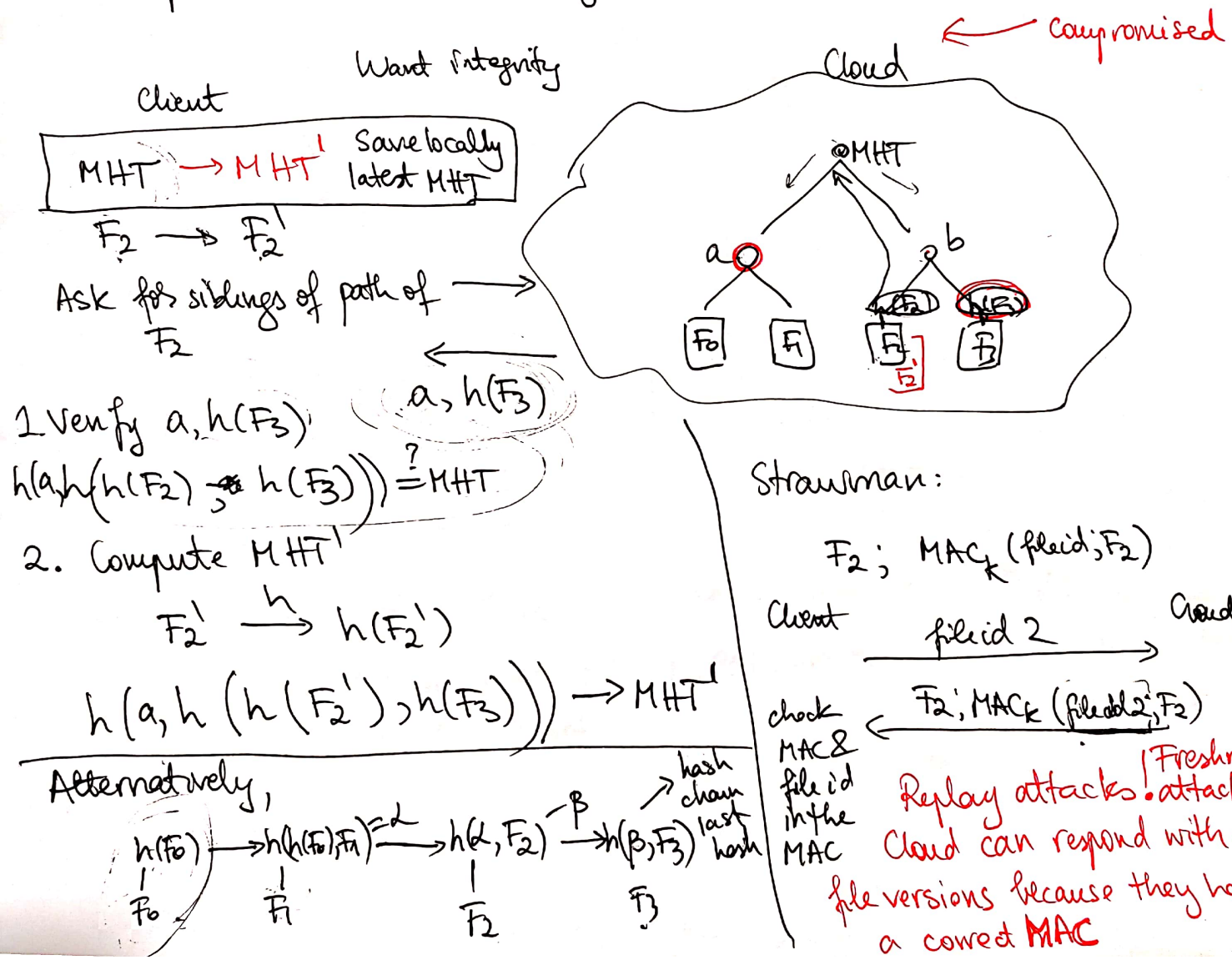
Assume that

$d_1' \neq d_1 \Rightarrow \underbrace{h(d_1')}_{b'} \neq \underbrace{h(d_1)}_{b}$

$h(a', b') \neq h(a, b) \Rightarrow i' \neq i$

$h(i', j') \neq \underbrace{h(i, j)}_{\text{MHT}}$

Example use case: secure storage



Strawman:

F_2; MAC_K(fileid, F_2)

Client → Cloud: fileid 2

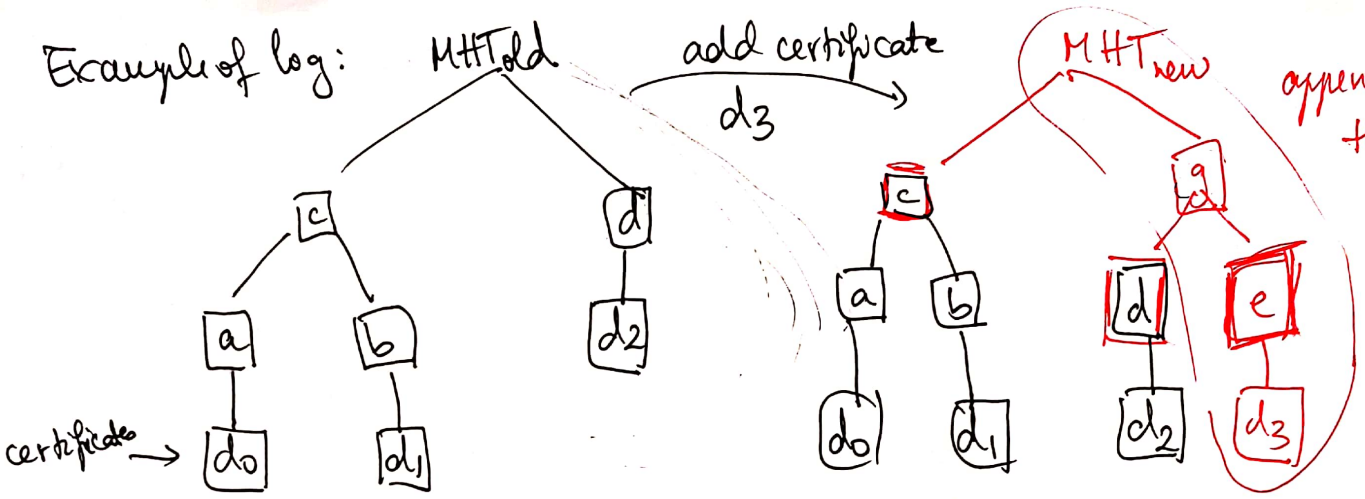
Cloud → Client: F_2', MAC_K(fileid 2, F_2)

check MAC & file id in the MAC

Replay attacks! Freshness attack

Cloud can respond with old file versions because they had a correct MAC

Example of log: MHT_{old} add certificate d_3 MHT_{new} append on the right



Append-only Merkle tree

- every certificate put in the log will stay there forever

Logs output MHT every period to all auditors

Auditors check that MHT_{new} is the result of append to MHT_{old}

Auditor has MHT_{new} and MHT_{old}, asks log for nodes:

log responds with this:
- old tree: c, d → root
- new tree: c, new node with kids d, e → new root

Auditor checks: $h(c,d) \stackrel{?}{=} MHT_{old}$

$h(c, h(d,e)) \stackrel{?}{=} MHT_{new}$

Ensures that a log can never erase a certificate because an auditor would detect this.

Certificate Transparency (CT)

Problem: – 2011 diginotar signed fake certs for *.gogle.com
CAs can get compromised

– ledger [– hosted centrally (eg. Google) + high throughput, scalable, no proof of work
– decentralized security

Insight: transparency: CT enables anyone to detect if there is a fake cert for them; it does not prevent creation of fake certificates

3 parties: – logs: store certificates, all certs in the world ⇐ Google
– monitors
– auditors

Logs: each log is a Merkle hash tree over certificates