Title need to change

Advicer : Gwan-Hwan Hwang
Student : Wei-Chih Chien

NTNU CSIE CCLAB

2015.10.29

Outline

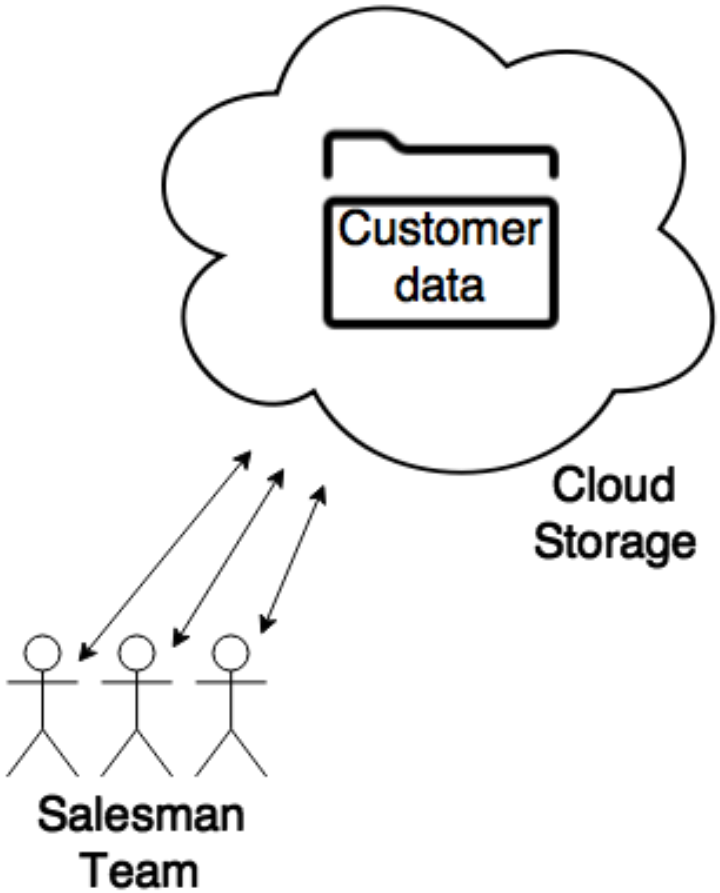
1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

Outline

1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

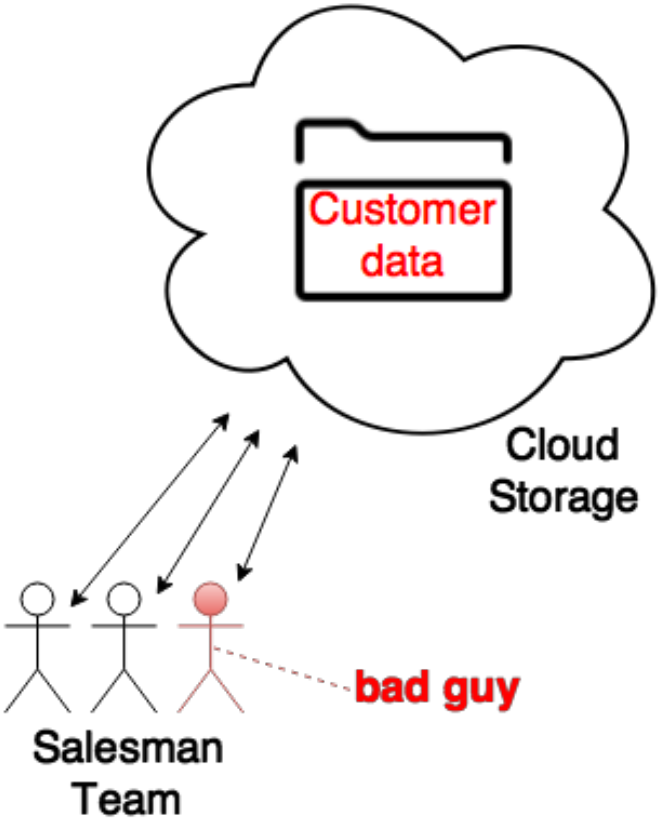
Scenario

Why Real-time Auditing?



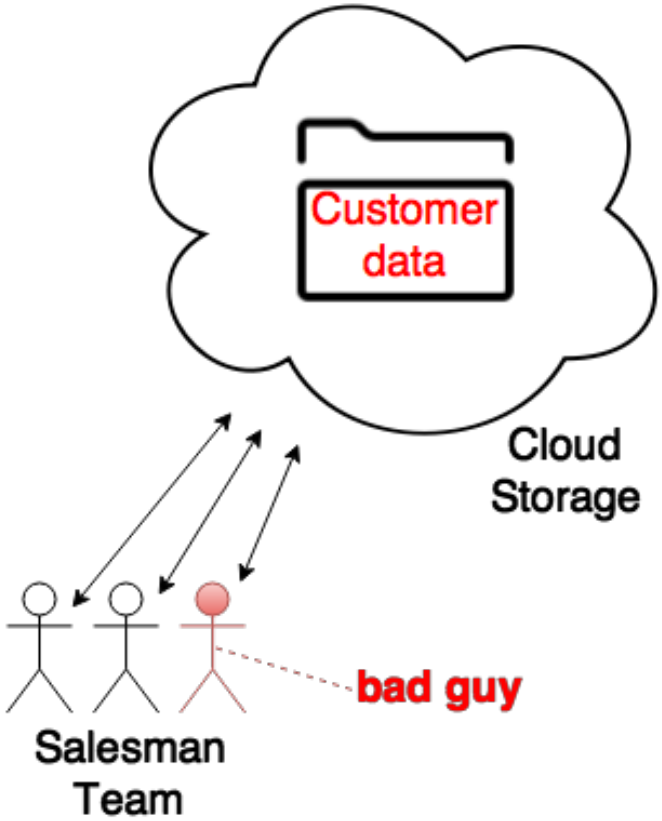
Scenario (CON'T)

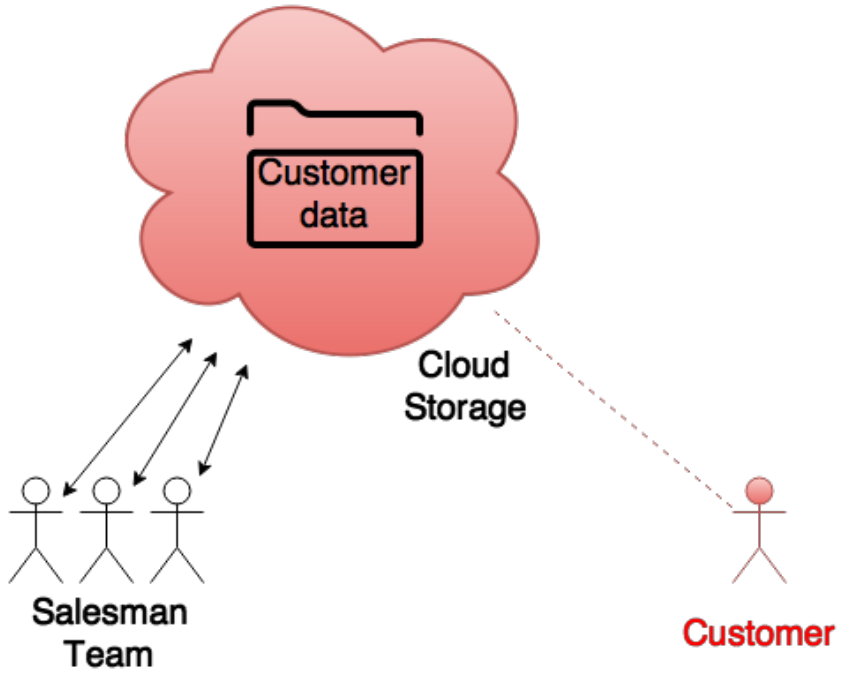
Why Real-time Auditing?



Scenario (CON'T)

Why Real-time Auditing?

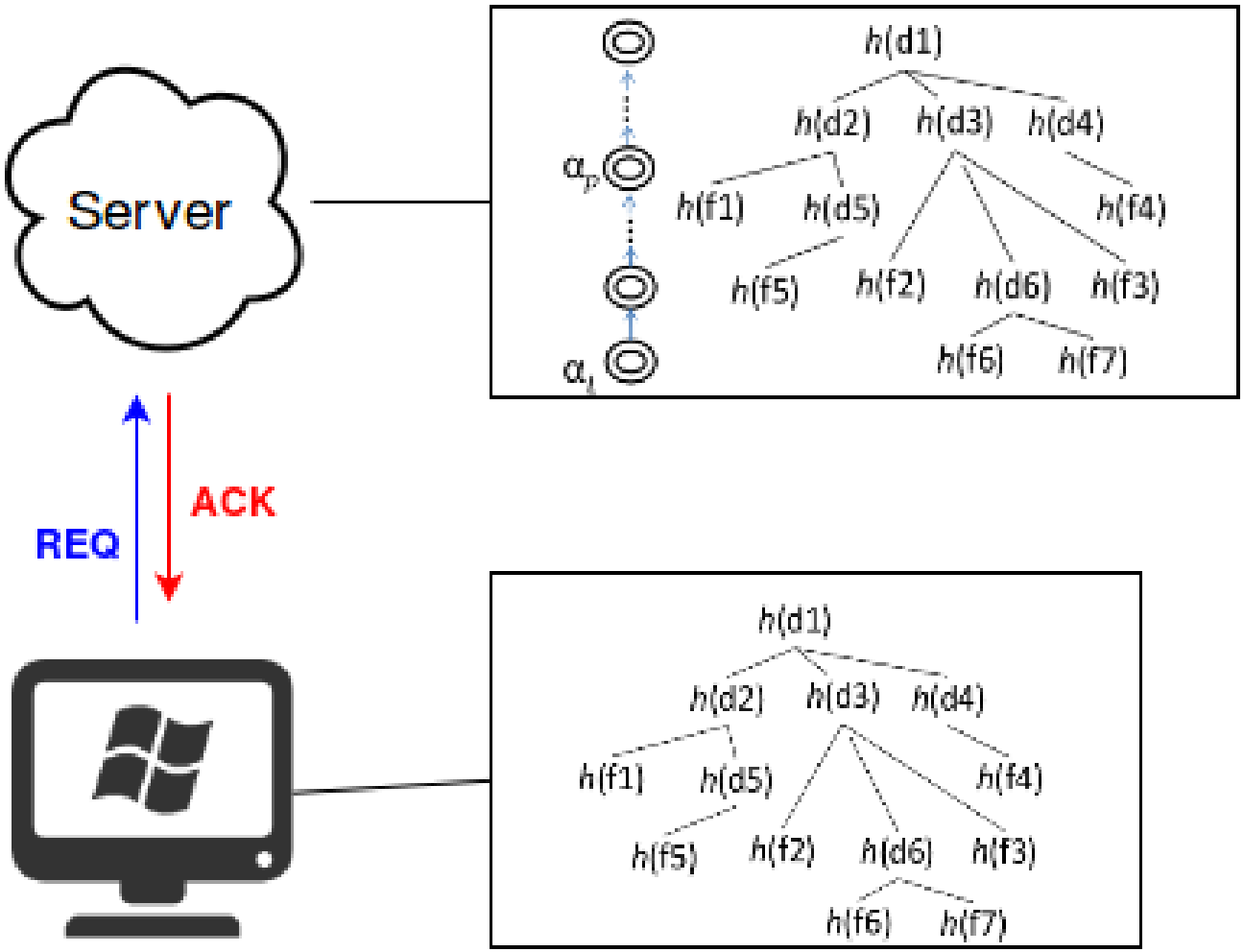


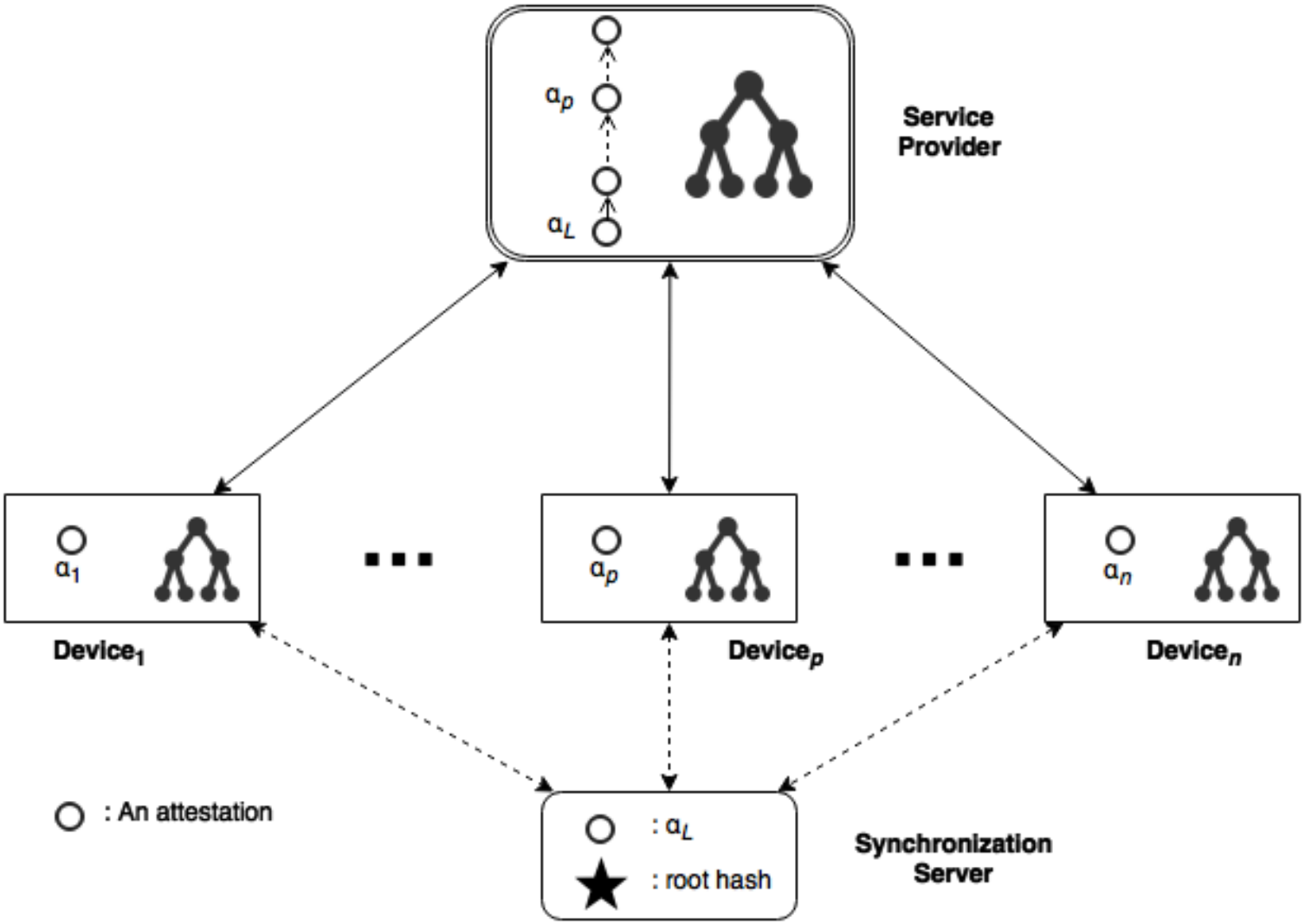


Outline

1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

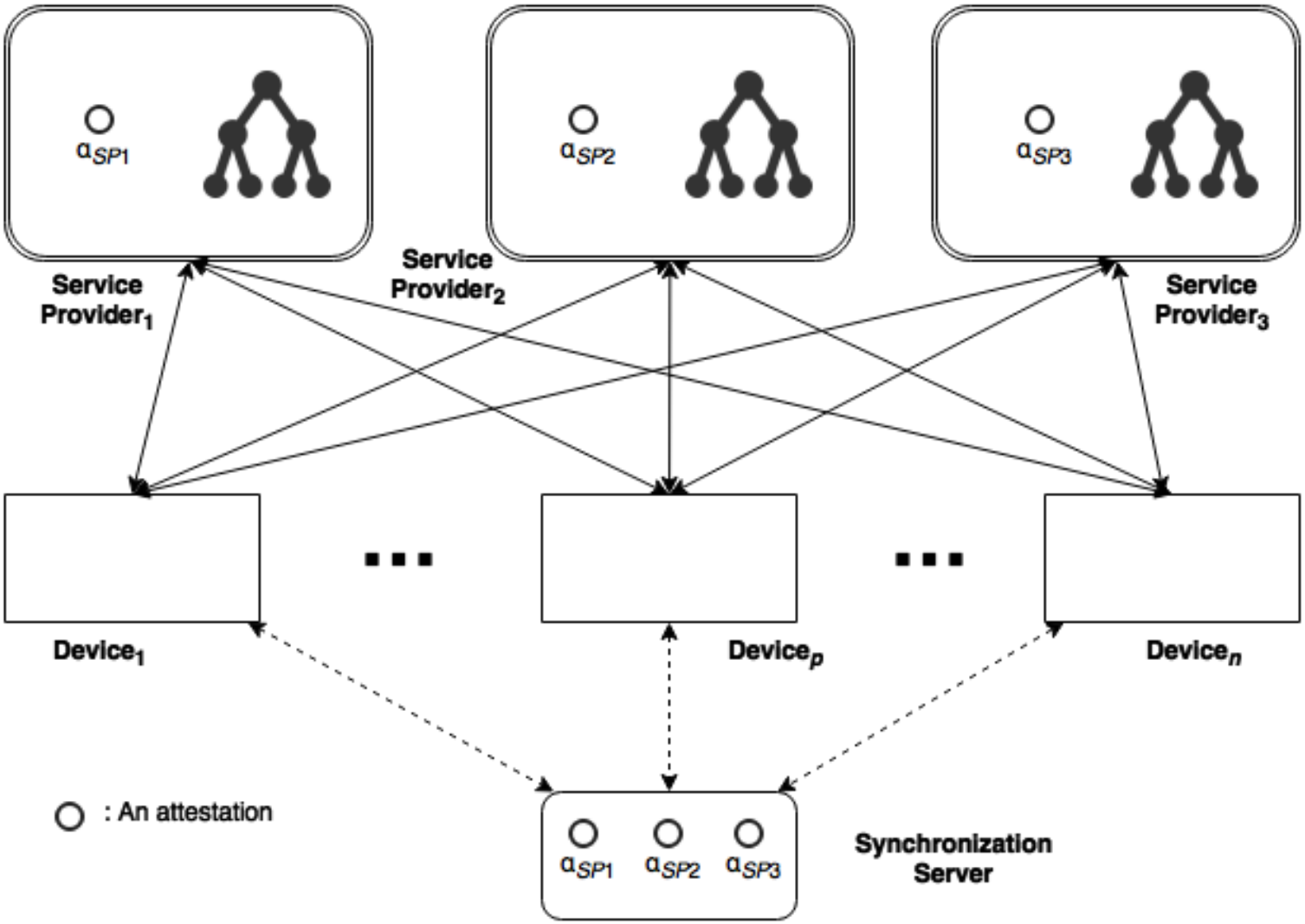
Intuitive Method





Worst-case : 累積大量未更新的動作造成系統緩慢

My Method



Assumption: 同時有k個server上同一file出問題的機率 ≈ 0

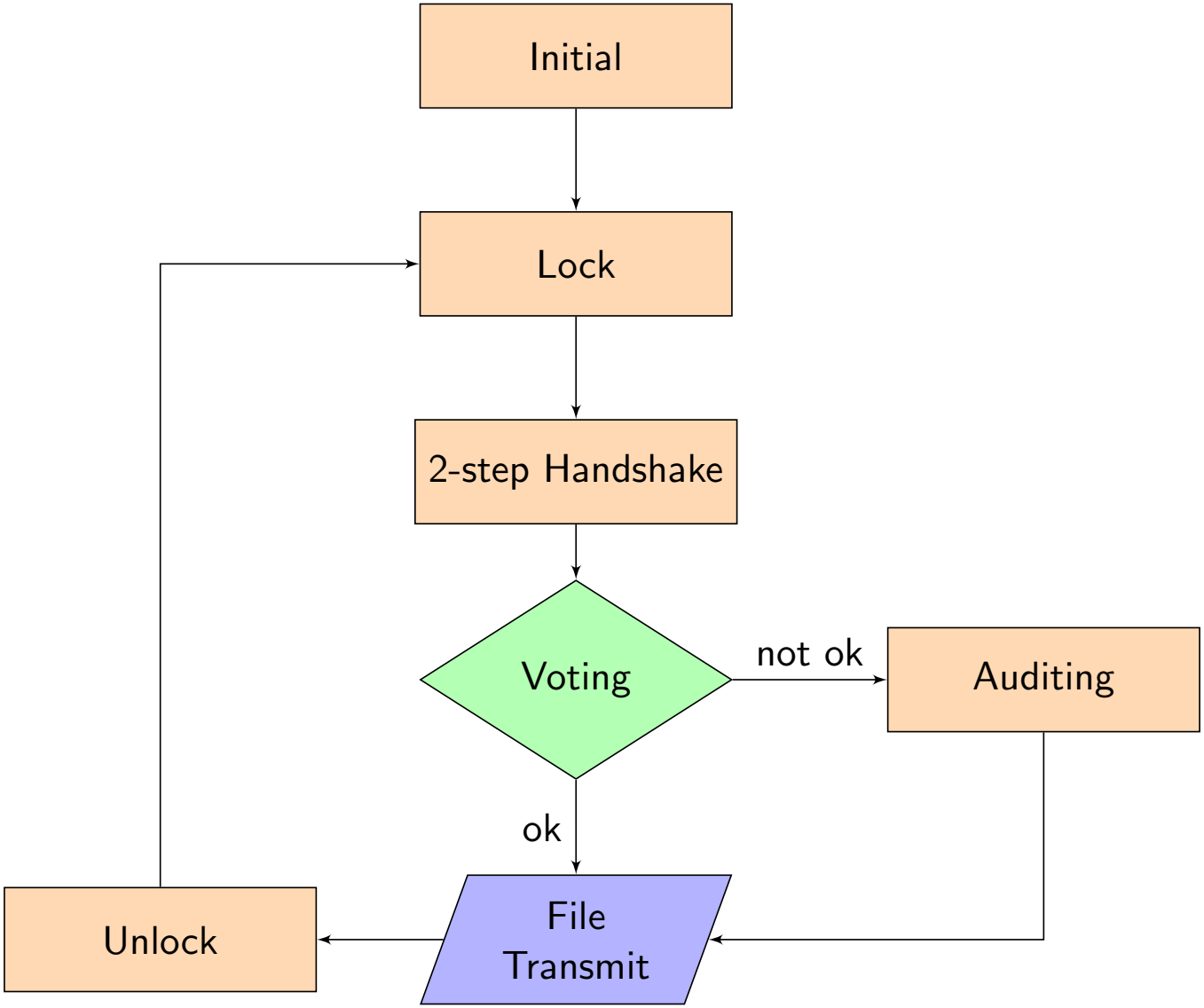
Comparison

- Pros
  1. Service Provider 不用累積證據
  2. Client 不用佔用空間儲存證據
  3. 資料有多份備份
  4. 花費較少的時間更新到最新的狀態
- Cons
  1. 硬體成本較高
  2. 需要處理多份 Response

Outline

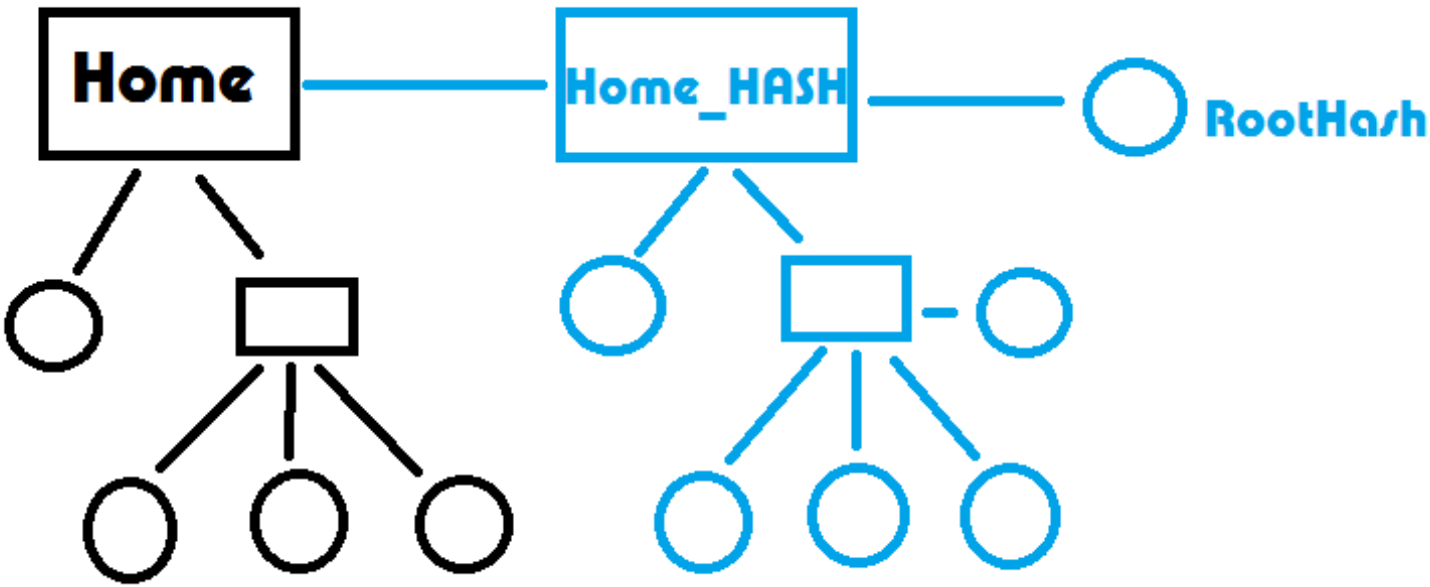
1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

Flowchart


Initial
Lock
2-step Handshake
Voting
not ok
Auditing
ok
File
Transmit
Unlock

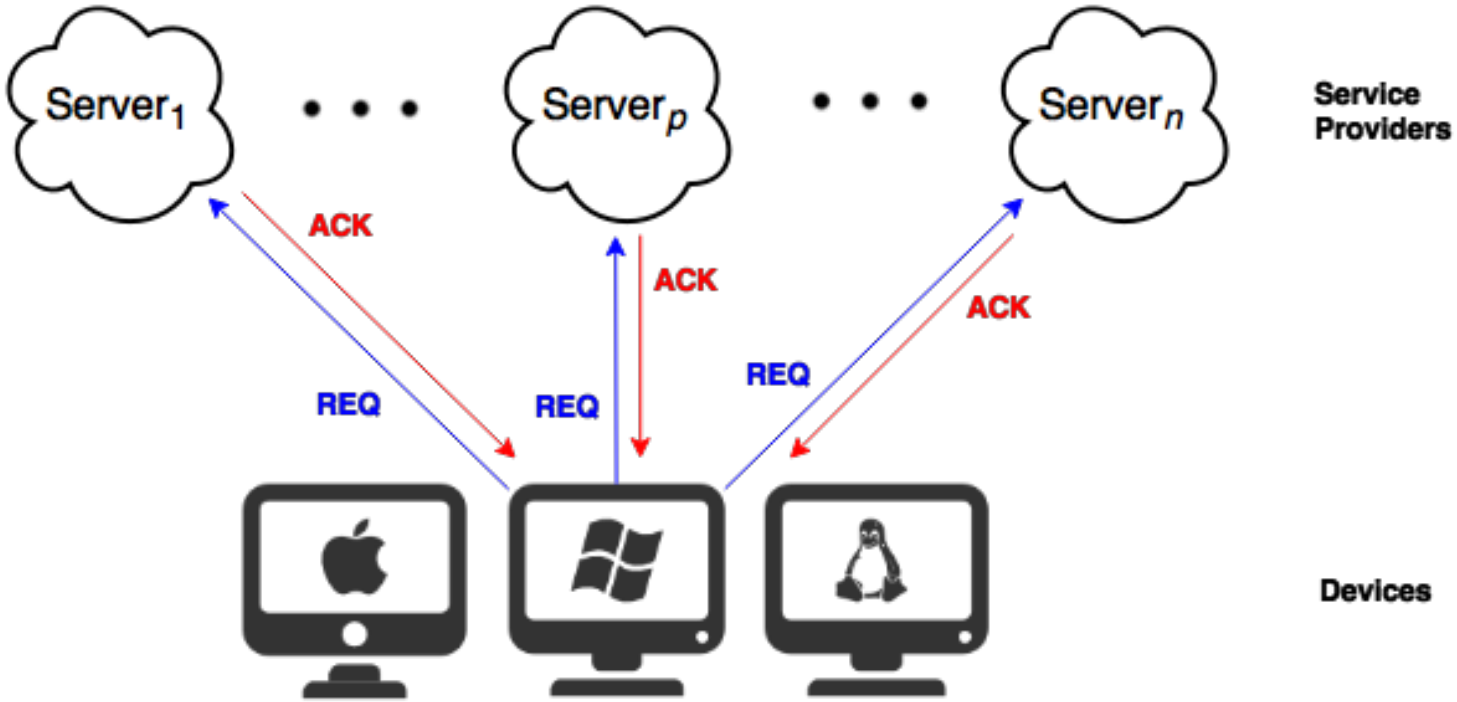
Initial

File → Merkle Tree



READ

I. 2-step Handshake & Voting

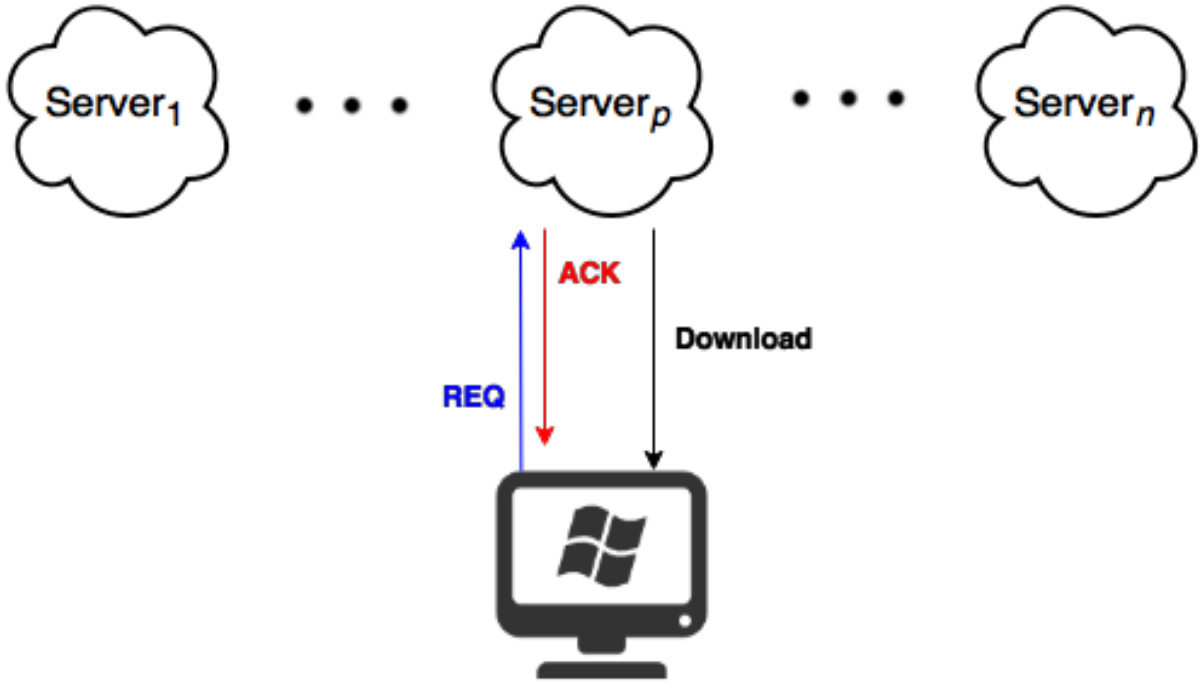


$$REQ = (op, [op]_{pri(D)})$$

$$ACK = (result, REQ, [result, REQ]_{pri(S)})$$

READ

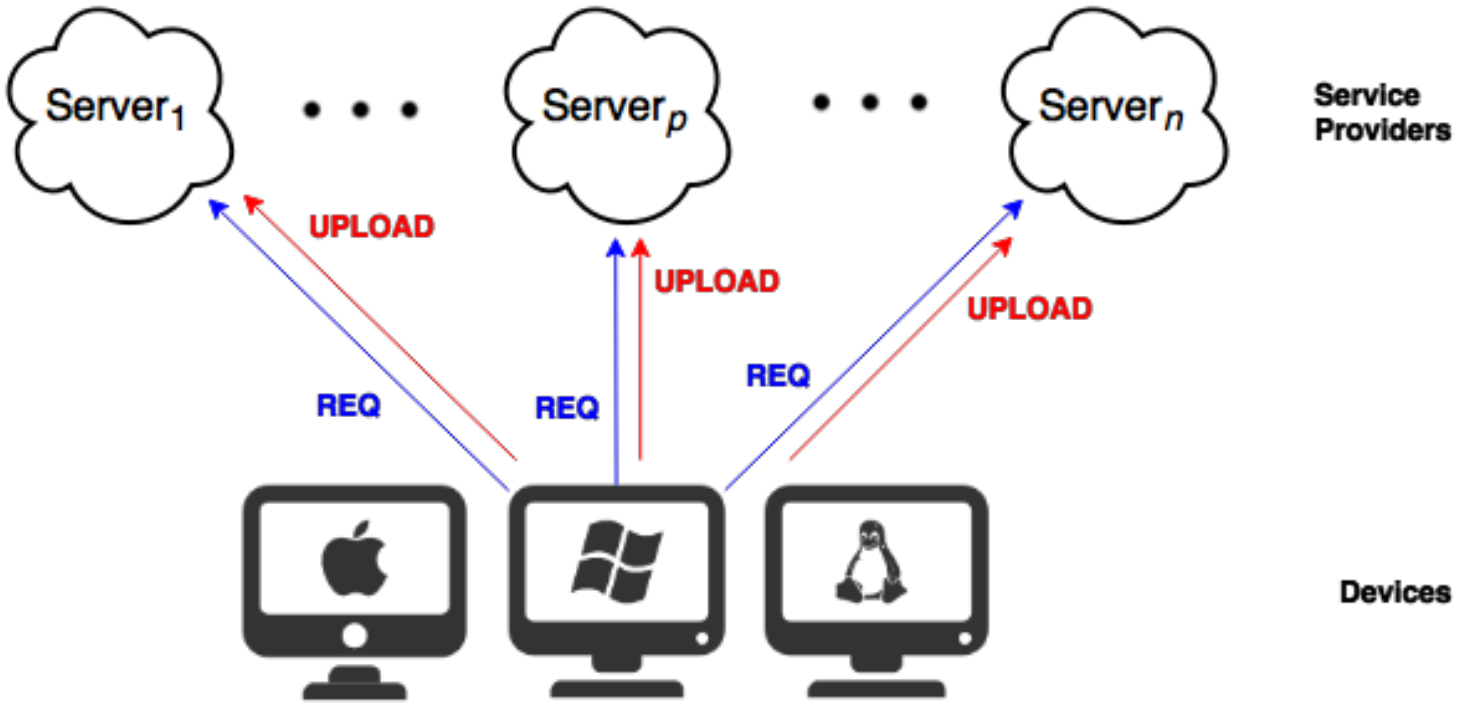
II. Download



$$\mathbf{REQ} = (\mathbf{op}, [\mathbf{op}]_{\mathbf{pri(D)}})$$
$$\mathbf{ACK} = (\mathbf{result}, \mathbf{REQ}, [\mathbf{result}, \mathbf{REQ}]_{\mathbf{pri(S)}})$$

WRITE

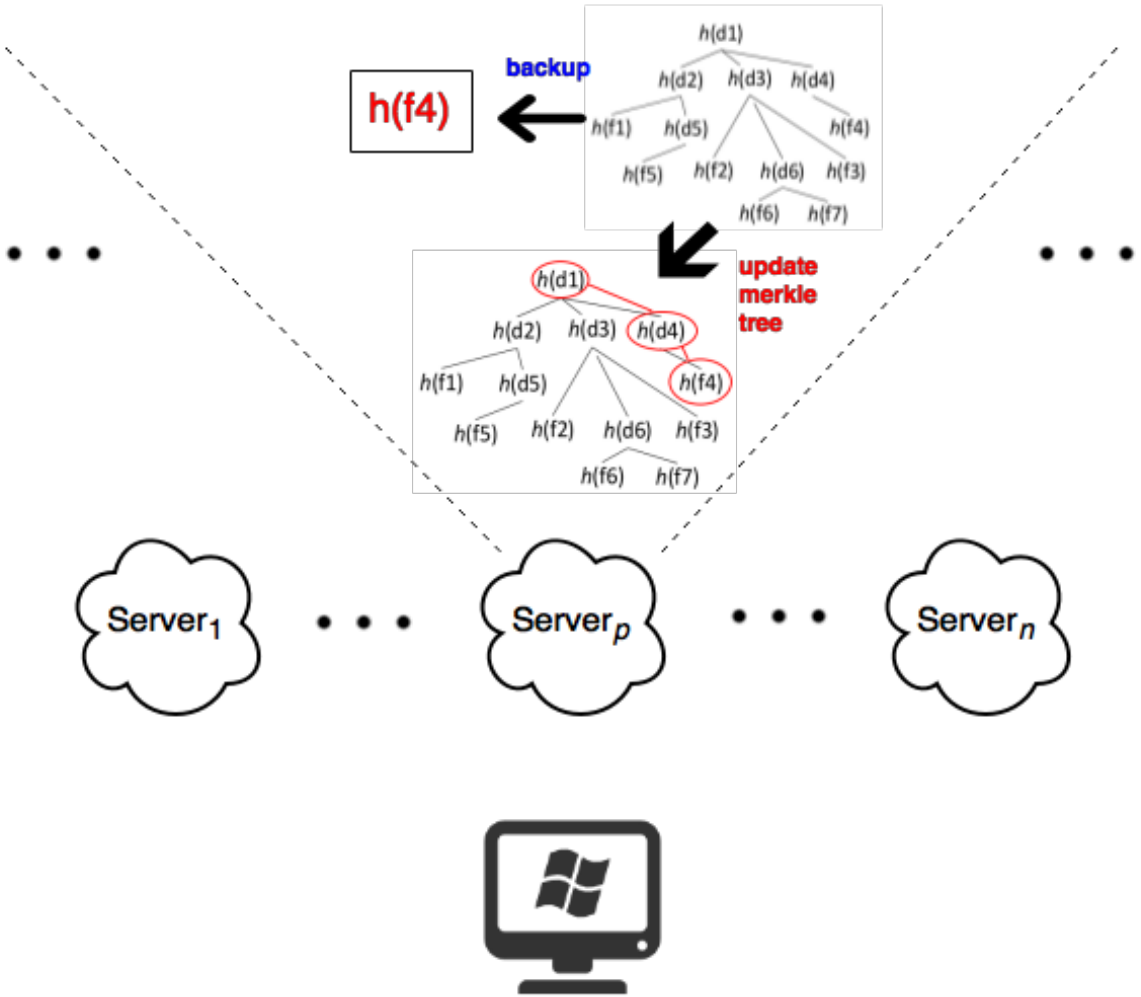
I. Upload



$$REQ = (op, [op]_{pri(D)})$$

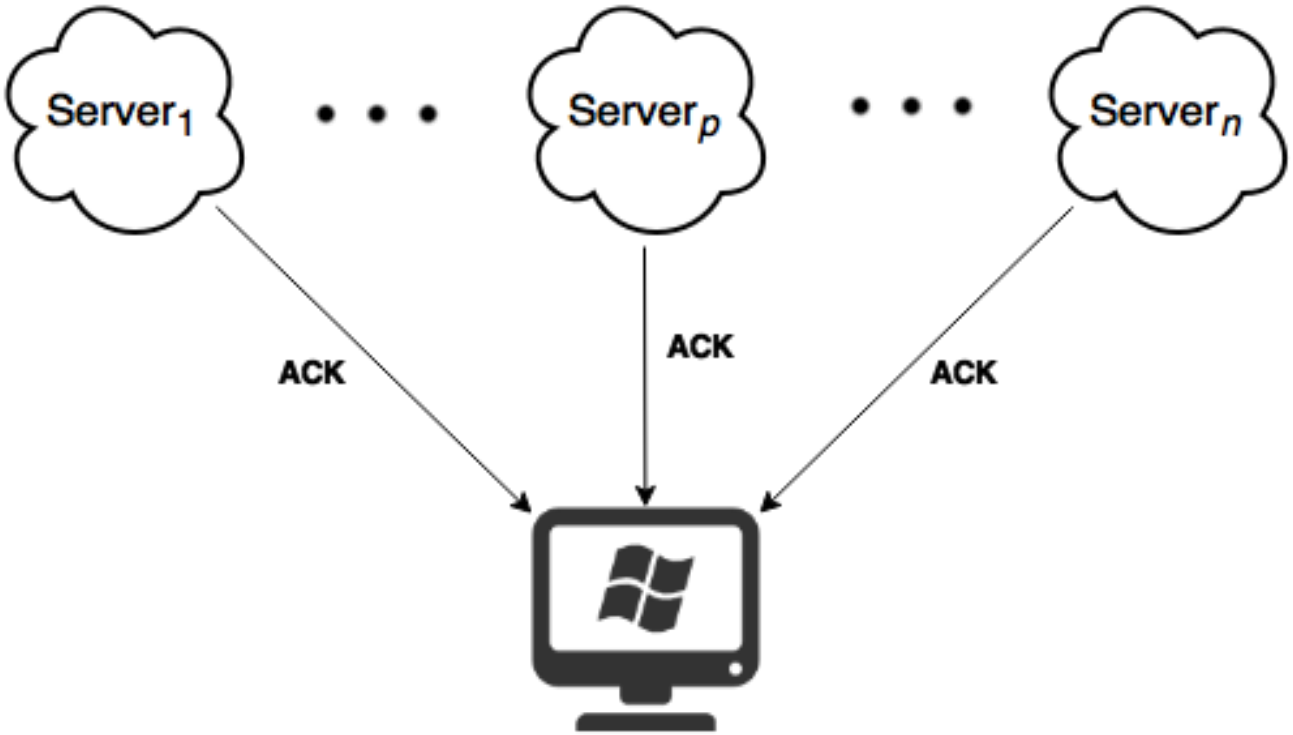
WRITE

II. Update Merkle Tree



WRITE

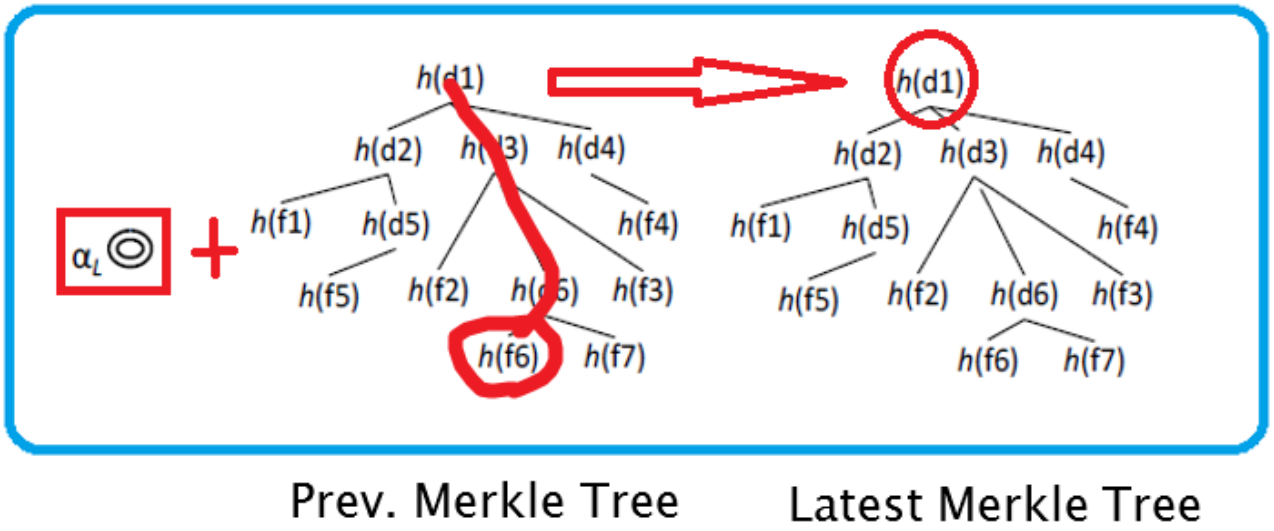
III. Voting



$$ACK = (result, REQ, [result, REQ]_{pri(S)})$$

AUDIT

1. Device 向 Synchronization Server 取得 Latest Ack.
2. Device 再向 Service Provider 取得 前一版本的 Merkle Tree.
3. 使用 Step I. 的 Ack 包含的檔案 Hash 值
   來更新 Step II. 的 Merkle Tree.



4. 比較 Device 自己算出的 Roothash 值是否和 Server 提供的相同.

Outline

1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

Create Merkle Tree

Do it again.

Account A	666 MB	42 files	6 directories
Account B	34 MB	54192 files	188 directories
Account C	6.54 GB	58484 files	1718 directories
Account D	20.6 GB	175389 files	5154 directories

Table : TIMES REQUIRED TO GENERATE THE ROOT HASH FROM NOT-HASHED FILES (IN SECONDS)

Account	Senior	My	MerkleTree Size
A	*3.404*	*3.645*	3.74 KB
B	*16.618*	*7.669*	3.77 MB
C	*229.351*	*242.198*	4.30 MB
D		*815.408*	12.9 MB

Operation Processing Time

Table : DOWNLOAD TIME (ms)

Account	100 times	Audit*
A	*4635*	*34 + 0*
B	*4660*	*33 + 0*
C	*5429*	*31 + 0*
D	*5554*	*31 + 0*

Table : UPLOAD TIME (ms)

Account	100 times	Audit*
A	*4322*	*41 + 7*
B	*5643*	*421 + 997*
C	*9236*	*421 + 2621*
D	*11466*	*1263 + 8085*

* download attestations time + audit time

Outline

1. Scenario
2. Real-time Auditing Schemes
   - Intuitive Method
   - Instant Auditing of Cloud Storage Access by Cache Partial Merkle tree
   - My Method
3. Protocol Detail
   - Flowchart
   - Initial
   - Read
   - Write
   - Audit
4. Experimental Results
5. Schedules

Schedules

1. My Method Finished.
   - Merkle Tree Implements.
   - Operation Handle (Read, Write and Audit).
   - File Transmit.
   - Object Transmit (Serialization).
   - Synchronization Server Implements.
2. Wei-Shian's Method Finished.
   - Attestation Chain Implements.
3. Design Different Experiments.

Thank You!