Verifiable Computation and Zero Knowledge Proof

Cloud Computing

Universities

Companies

Individuals

Government

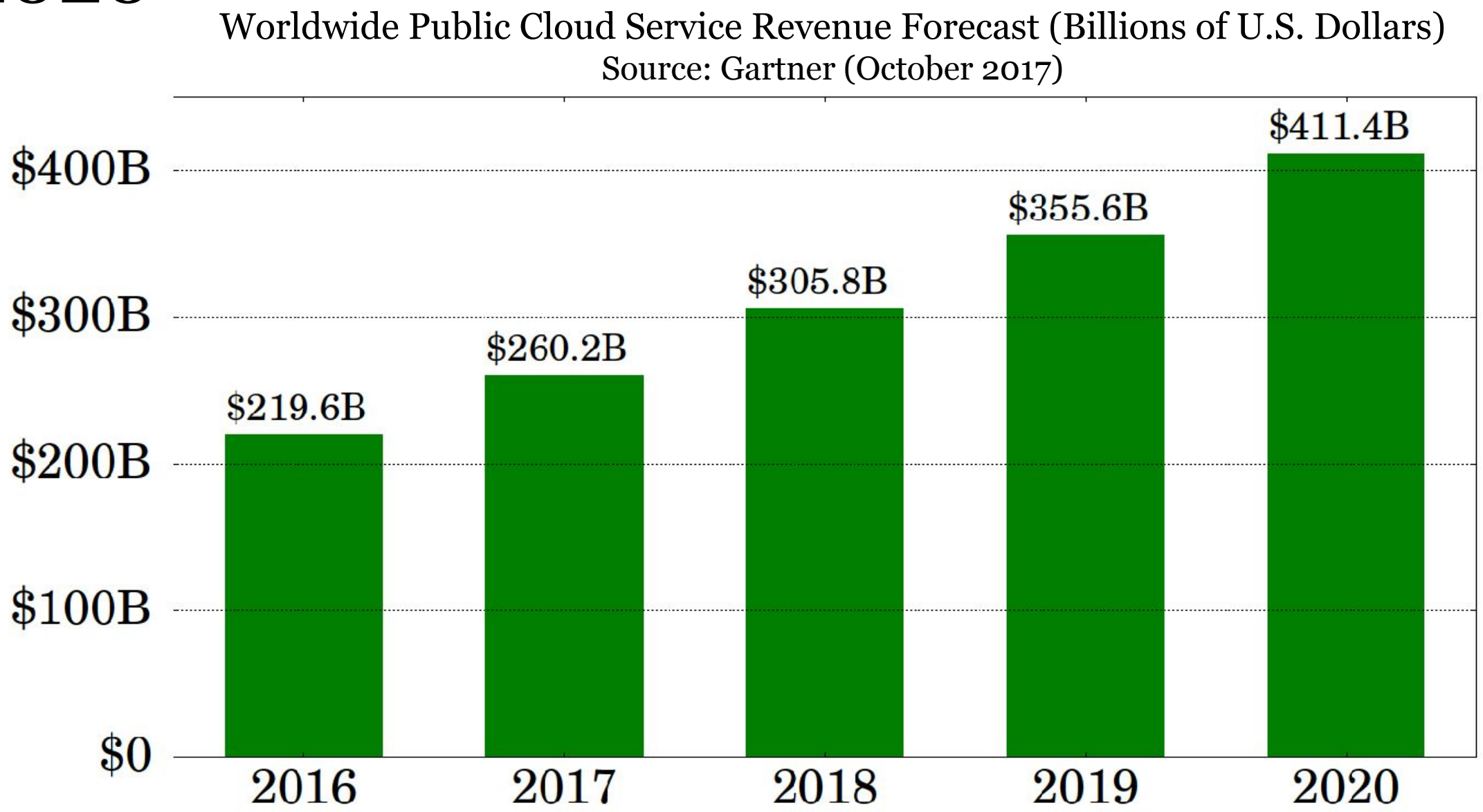
DOD looks to get aggressive about cloud adoption

By Derek B. Johnson | Sep 20, 2017

Benefits:

- Reduce local computa
- Elasticity and geograp

Cloud Computing Market to Reach $411B by 2020



Security Issue: Integrity

Universities

Companies

Individuals

Government

Server failure
Software error
Data tampering in transmission
Random answer
Malicious server behavior
...

Data Can be Lost or Corrupted

Amazon web service suffers major outage, disrupts East Coast internet

Published February 28, 2017 · Fox News

Illustration file picture (REUTERS/Kacper Pempel/Files).

If you are experiencing a sluggish web browser then it could be the result of an Amazon web service outage.

An outage hit Amazon Web Services Tuesday, reportedly impacting lots of web pages. Specifically, the cloud giant is experiencing problems with its Simple Storage Service (S3) on the East Coast. Widely used for backup and archive, S3 is harnessed by a host of companies.

Corrupt iCloud Data Causes iOS SpringBoard Home Screen Crash (with Fix!)

By Dave Hamilton Apr 15th, 2013 5:

In addition to 3rd party app data, iCloud also stores data for a few Apple apps and services, and if this data gets corrupted you can wind up with a very unstable iPhone (or iPod or iPad). Unfortunately Apple doesn't provide a way to let you tell iCloud to reset this data. Short of permanently disabling "Documents & Data" syncing with iCloud, the only way to fix this is to delete the offending data from the [home]/Library/Mobile Documents folder on your Mac and reboot your iOS device.

The Full Story

A few weeks ago I began experiencing a recurring Crash on our iPod touch here. The is

How to guarantee the integrity of data and computations

Verifiable Computation (VC)

server

client

function f

result = f(data) + 

digest δ

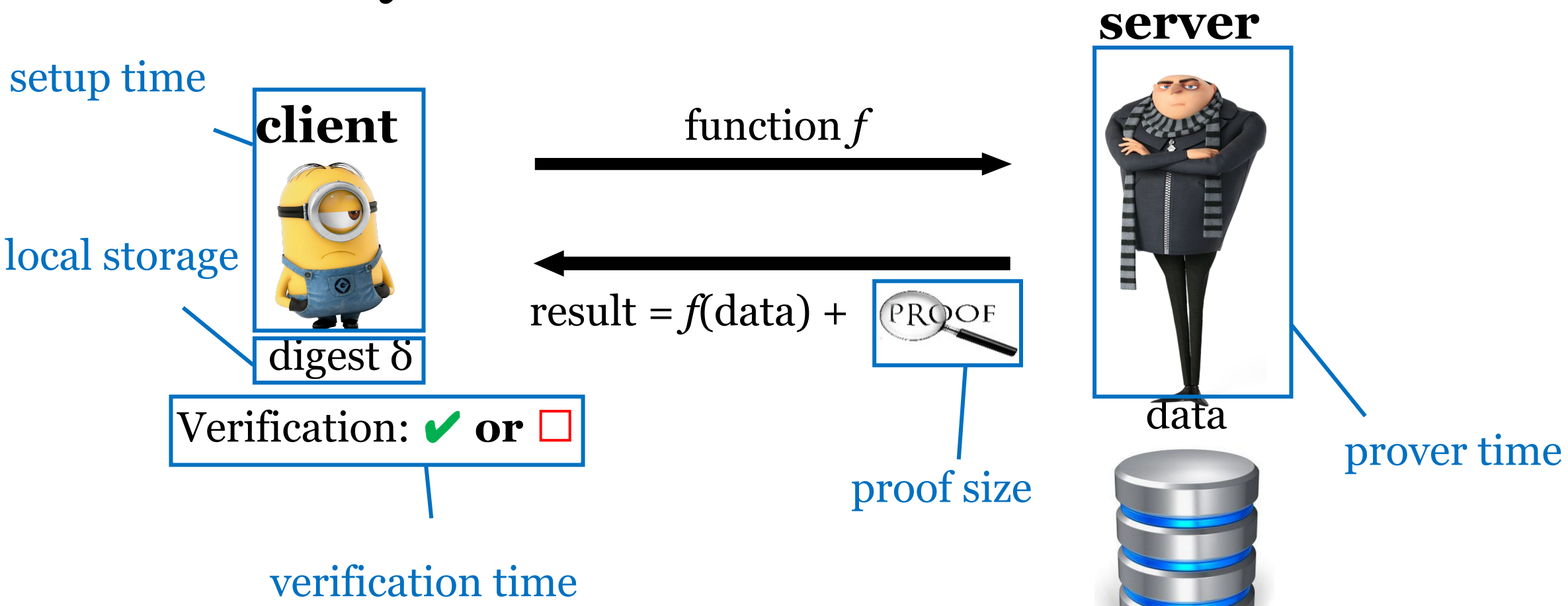
data

Verification: ✔ **or** ☐

Correctness/completeness: Pr[result = f(data) and proof is honest and verification is ✓] = 1

Soundness/security: Pr [result ≠ f (data) and verification is ✓] ≤ $\frac{1}{2^{100}}$

Efficiency measures



Requirements:

- Local storage < data
- Proof size < data
- Verification time < f

Customized schemes vs generic schemes

server

client

digest δ

function f

result = f(data) + 

data

Verification: ✔ **or** ☐

Customized schemes

× Only support limited operations

✔ Efficient

Authenticated data structures

Generic schemes

✔ Supports all functions, modeled as arithmetic circuits

× Slow setup time and prover time

Merkle hash tree

- Cryptographic hash:
  - $H: \{0,1\}^* \rightarrow \{0,1\}^k$ any string to 256-bit string, deterministic
  - Collision resistant: hard to find x,y such that H(x) = H(y)

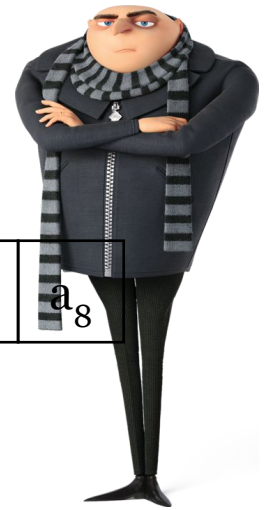
- Merkle tree: integrity of elements in a vector

Integrity of storage

server

client

3^{rd} element



a_3

result

a_1	a_2	a_4	a_5	a_6	a_7	a_8

proof

digest = $H(a_1, a_2, \dots, a_8)$

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8

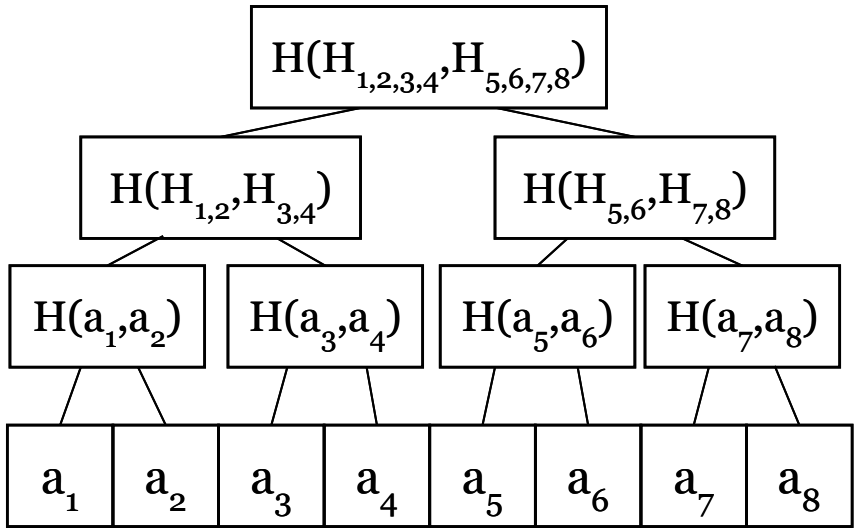
Verification: digest = $H(a_1, a_2, \dots, a_8)$

Merkle hash tree

client

server

digest =root



Merkle hash tree

client

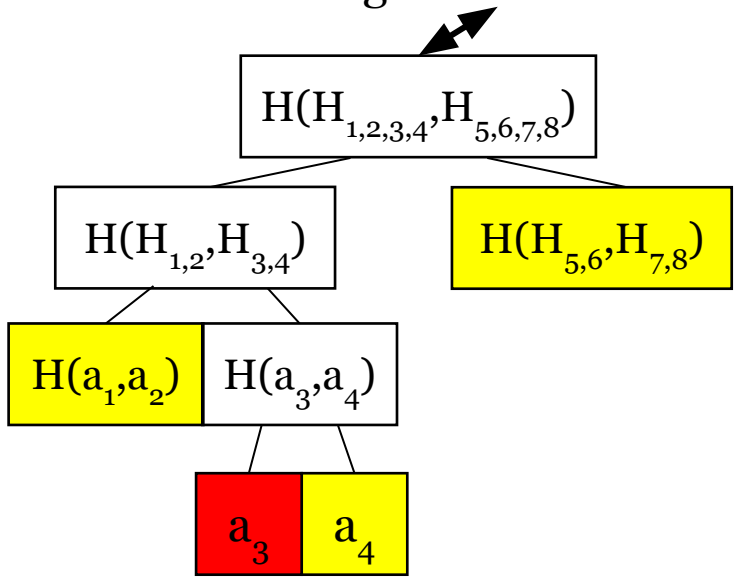
3rd element

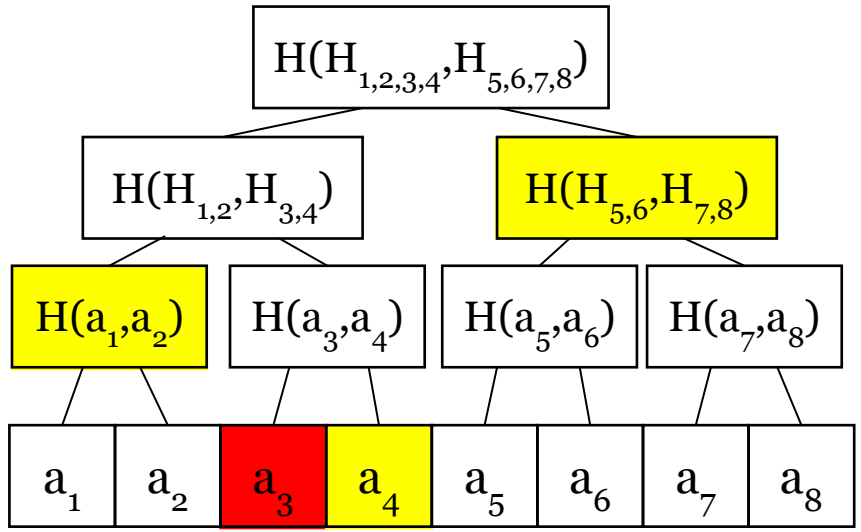
a_3 result

a_4 $H(a_1,a_2)$ $H(H_{5,6},H_{7,8})$ proof

server

digest =root

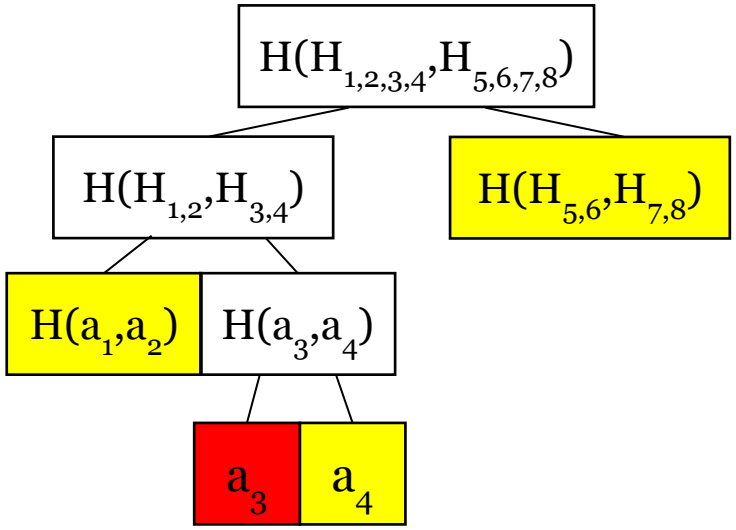


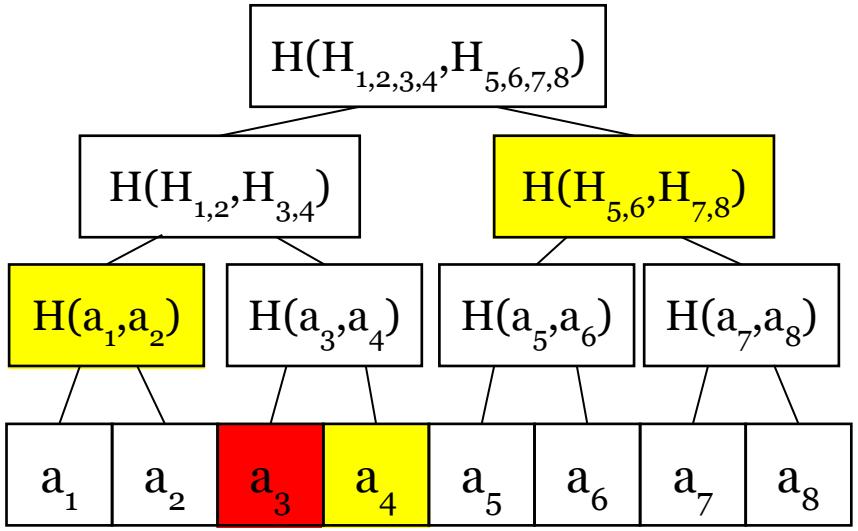


Merkle hash tree

- Correctness

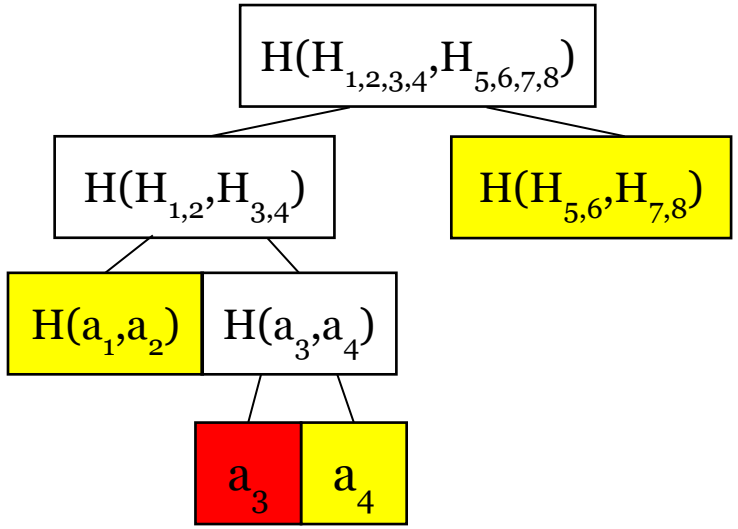
- Soundness: cheat → break collision resistant of hash function

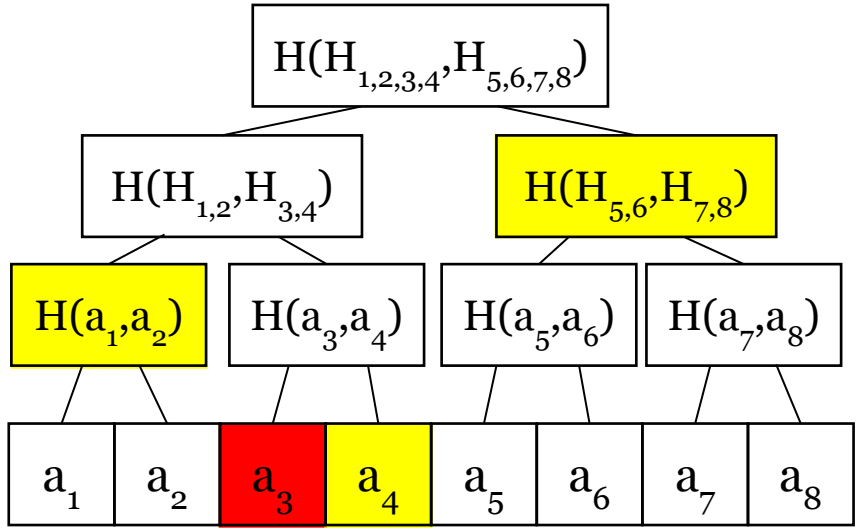




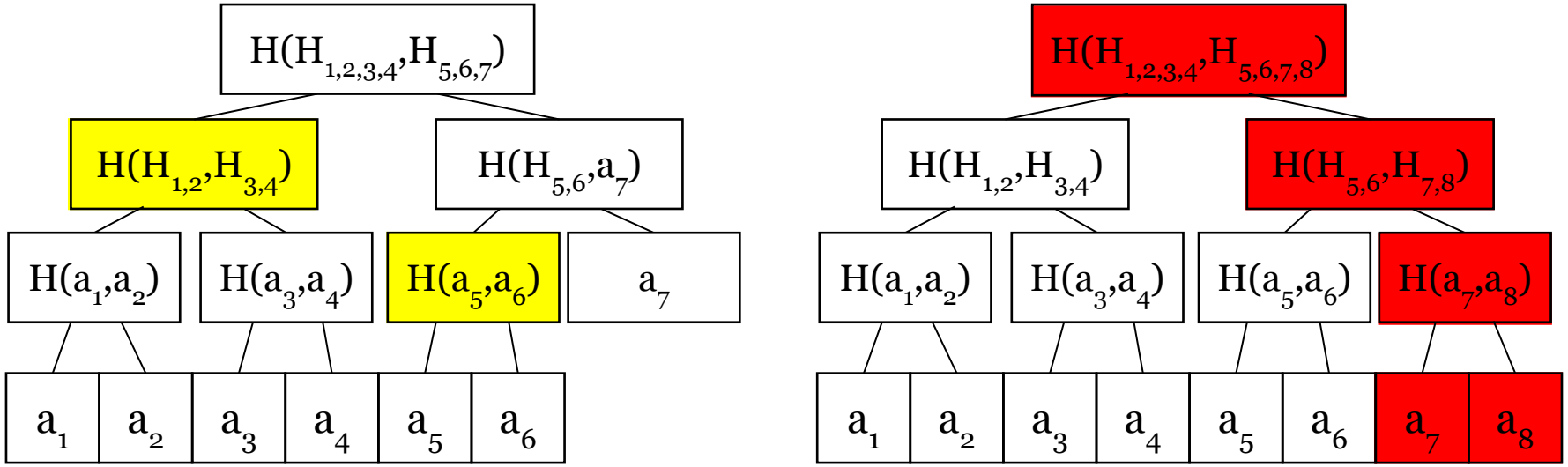
Complexity

- Local storage: O(1)
- Setup: O(n)
- Prover time: O(n) or O(log n) with O(n) storage
- Proof size: O(log n)
- Verification time: O(log n)





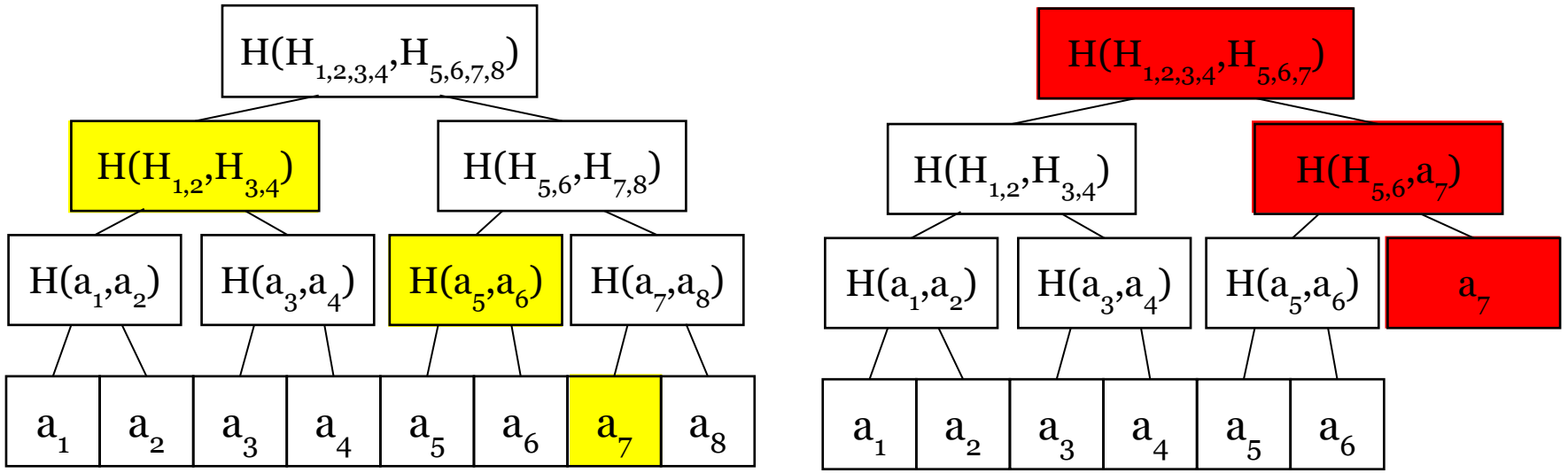
Update



Insertion:

1. P sends proof for a_7 , V validates it
2. V updates the root using a_8 and only the proof
3. P updates the root

Update



Deletion:

1. P sends proof for a_8 , V validates it
2. V updates the root using only the proof
3. P updates the root

Complexity

- Local storage: O(1)
- Setup: O(n)
- Prover time: O(n) or O(log n) with O(n) storage
- Proof size: O(log n)
- Verification time: O(log n)
- Update: O(log n)

Difference from digital signatures

- Digital signatures: sign with private key, everyone can validate with public key

- Sign (i, a_i) with digital signatures

- Local storage: O(1)
- Setup: O(n)
- Prover time: O(1)
- Proof size: O(1)
- Verification time: O(1)
- Update: add O(1), delete? O(n)

Other authenticated data structures

- Authenticated data structures
  - Authenticated skip list
  - Authenticated red-black tree
  - Authenticated dictionary

Topics

- Customized schemes: accumulators, set operations
  - RSA accumulator
  - Bilinear accumulator

- Generic schemes:
  - SNARKs
  - Interactive proof

Zero knowledge proof

verifier

$f(x,w) = y$ + 

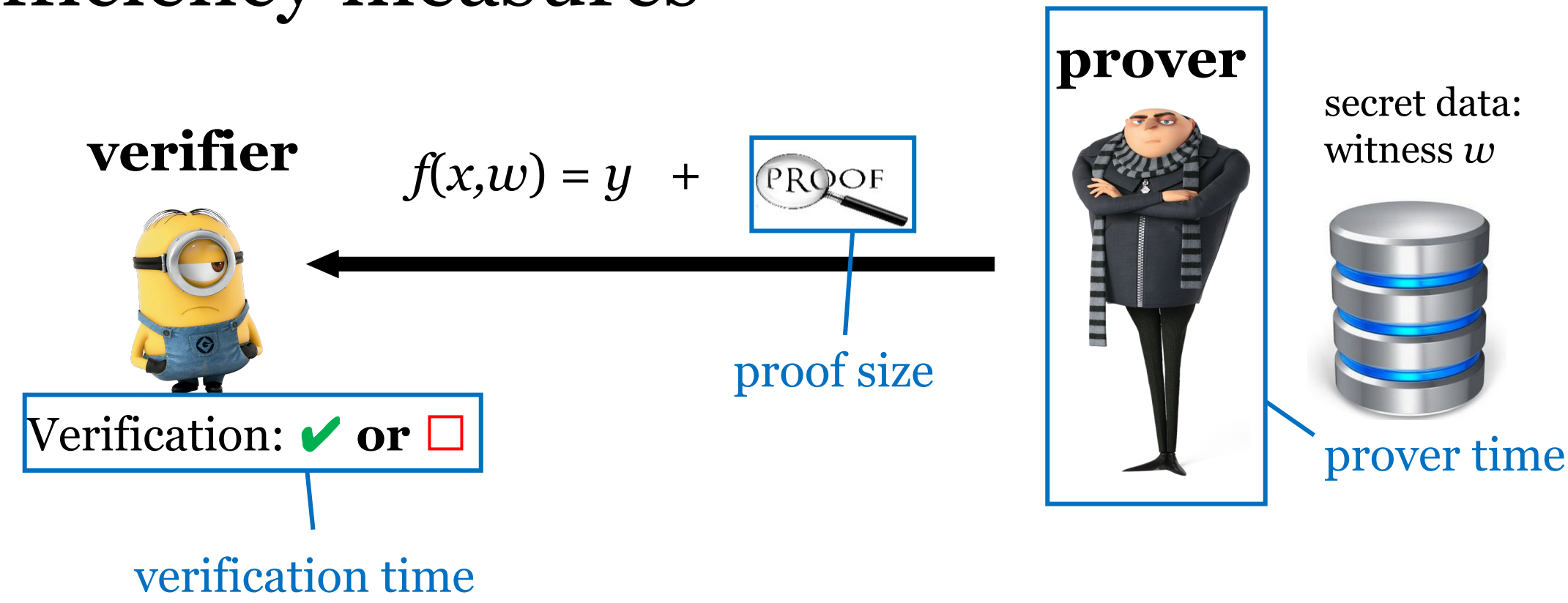
prover

secret data:
witness w

Verification: ✔ **or** ☐

- Completeness: Pr[honest prover and verification is ✓] = 1
- Soundness: Pr[$y \neq f(x,w)$ and verification is ✓] ≤ $\frac{1}{2^{100}}$
- Zero knowledge: proof leaks no information about w

Efficiency measures



Requirements?

Applications

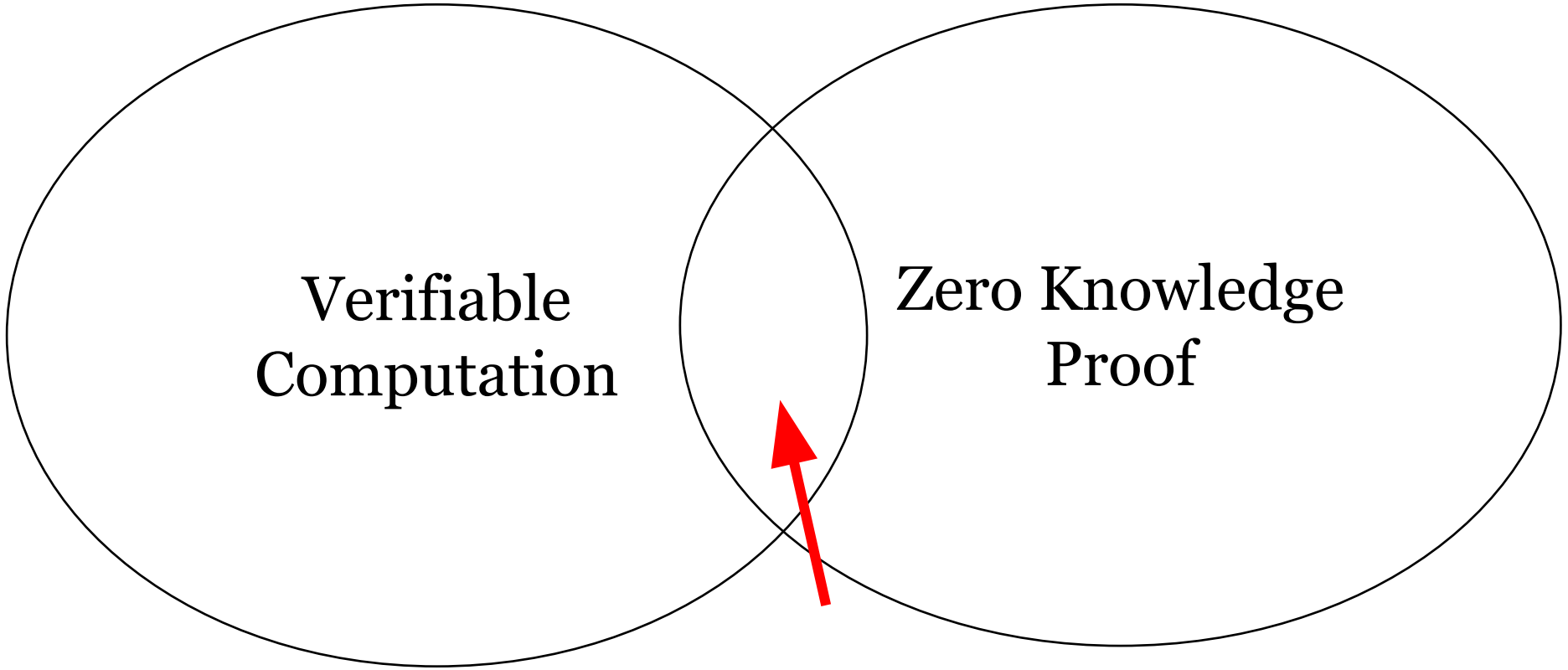
- Blockchain and cryptocurrencies
- Anonymous credentials

Relationship to secure multiparty computation (MPC)

- Malicious secure MPC can be used as zero knowledge proof with proof size and verification time $O(|C|)$

- Zero knowledge proof can be used to lift semi-honest MPC to malicious MPC
  - Weak requirement on proof size and verification time $O(|C|)$
  - Strong requirement on concrete efficiency of prover time

Relationship to verifiable computation



Best of both: succinct zero knowledge proof

- Proof size is smaller than *f*
- Verification time faster than computing *f* *