Consensus in blockchains: Overview and recent results

Christian Cachin
University of Bern

ViSP, 2023

Overview

UNIVERSITÄT BERN

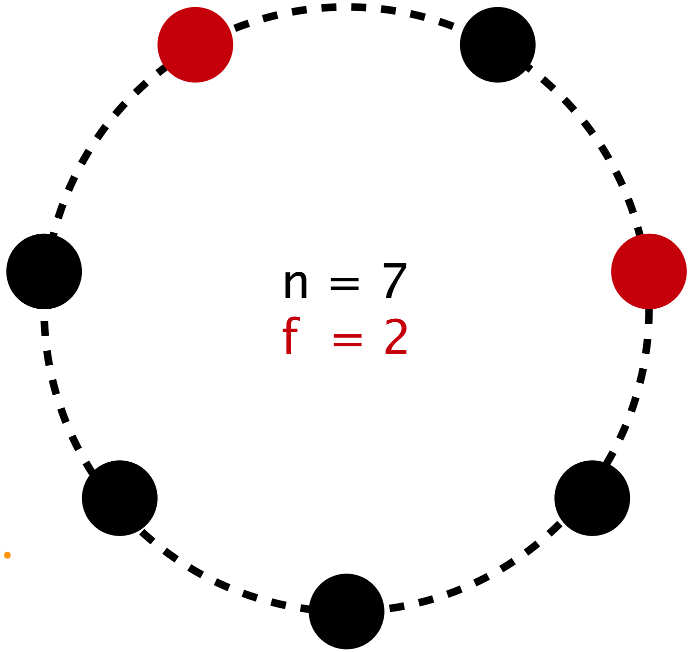
- **for** *model* $\in$ all kinds of blockchain consensus **do**
  - describe *model*

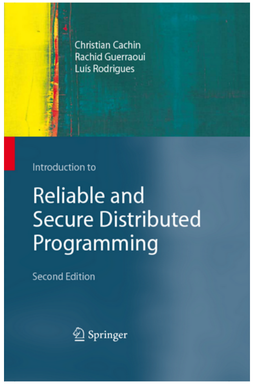
- **while** time lasts **do**
  - select some *result* $\in$ `https://crypto.unibe.ch/pub`
  - present *result*

- Answer your questions

1 – Threshold trust

- Trust by numbers
  - n nodes total
  - f faulty (Byzantine) nodes
- Homogeneous and symmetric
- Requires n > 3f
- Tendermint, DiemBFT, Quorum ...

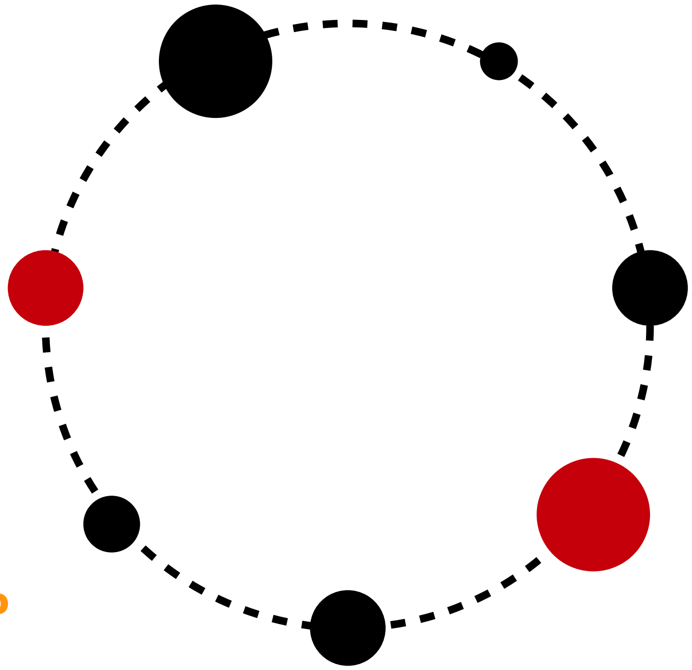




b
UNIVERSITÄT
BERN

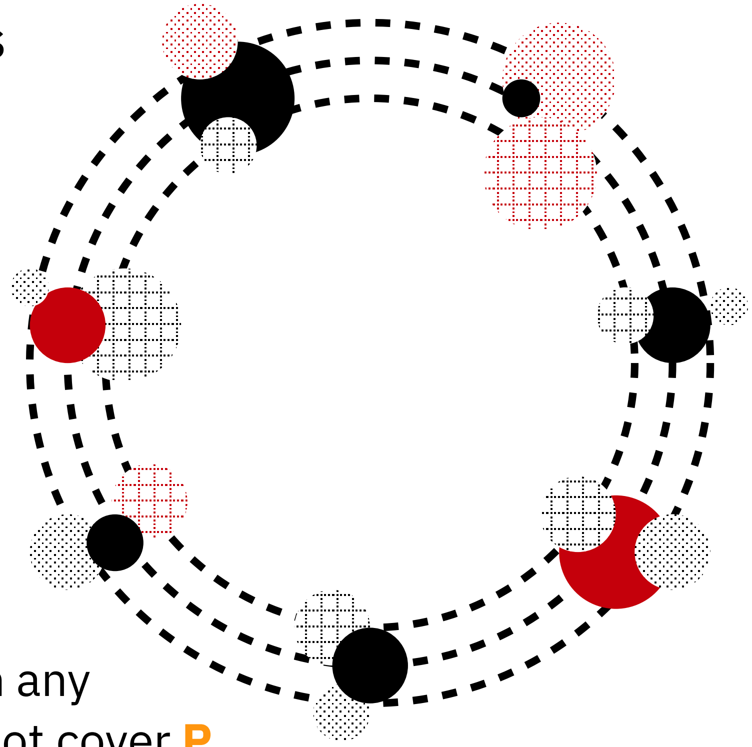
2 – Generalized trust

- Trust by generalized quorums
  - Set of nodes **P**
  - Fail-prone sets consisting of possibly Byzantine nodes
  - Byzantine quorum system
- Heterogeneous and symmetric
- Requires Q3-property
  - Any fail-prone sets must not cover **P**
- Not used by any cryptocurrency (!)



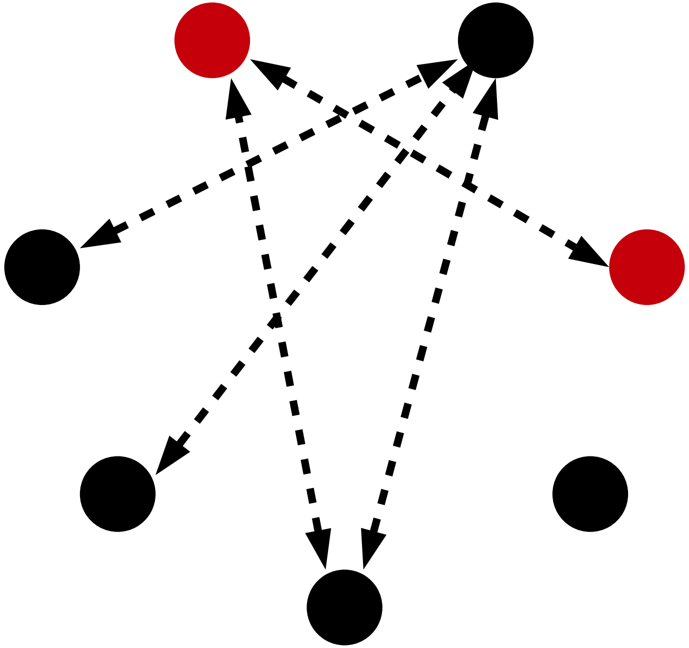
3 – Asymmetric trust

- Subjective generalized quorums
- Every node has its own Byz. quorum system on **P**
- Consistency across nodes' quorum systems
- Requires B3-property
  - ∀ p, p' : any fail-prone set of p with any set of p' and any set of both must not cover **P**
- Ripple, Stellar, [CT19]



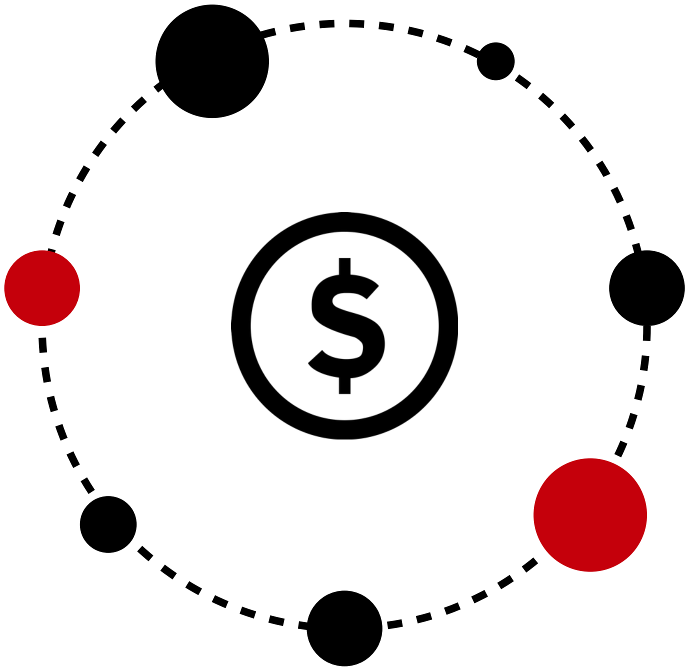
4 – Unstructured, probabilistic voting

- Random sampling of peers
- Exchange information and votes
- Usually coupled with a DAG on transactions
- Avalanche, Conflux, IOTA-Tangle



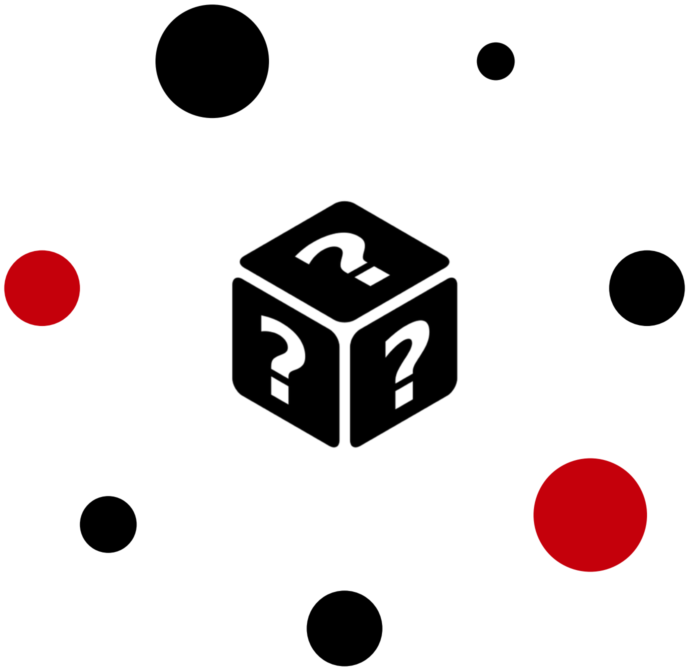
5 – Stake-based voting

- Stake determines voting power
- Protocols generalized from symmetric voting
- Cosmos, EOS, NEO, Aptos, SUI ...



6 – Stake-based probabilistic choice

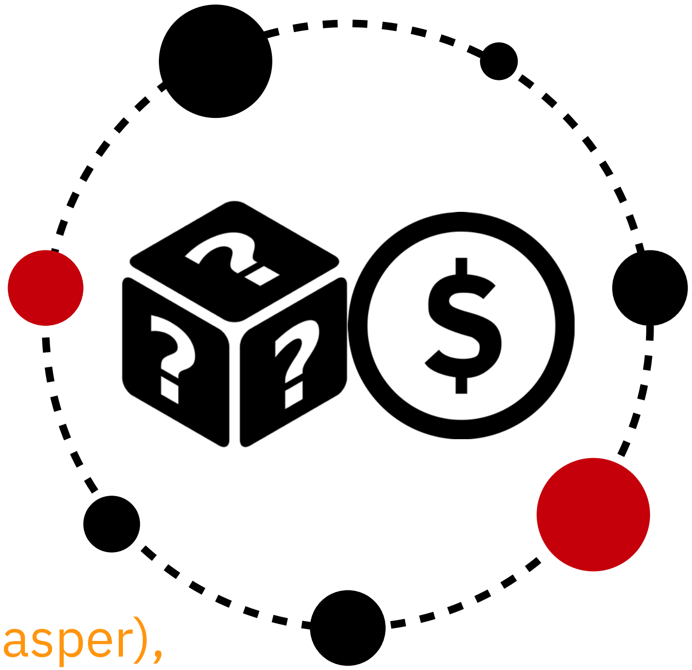
- Lottery according to stake
- Probabilistic leader election
- Cryptographic sortition using a verifiable random function (VRF)
- Cardano/Ouroboros ...



UNIVERSITÄT BERN

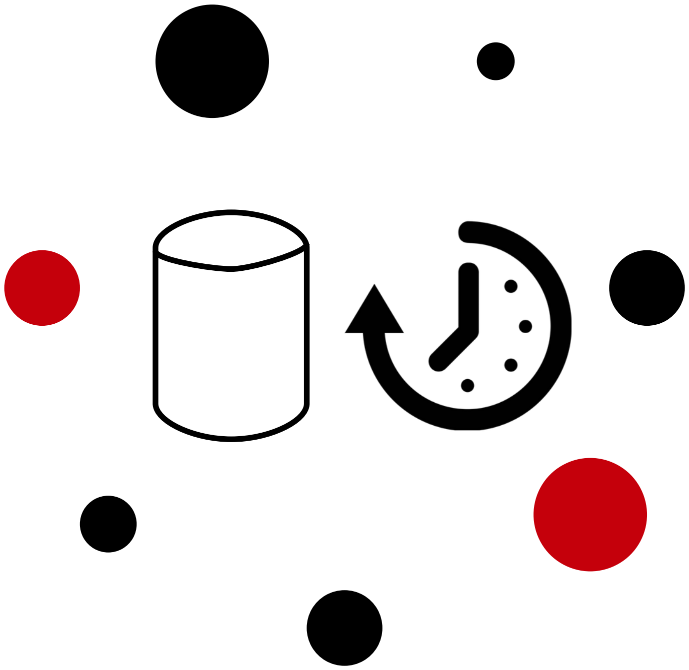
7 – Hybrid prob. choice and stake voting

- Stake determines probability or voting power
- Mix of random choice with voting
- Slashing of invested stake upon detection of misbehavior
- Ethereum (LMD-GHOST & FFG-Casper), Polkadot (BABE & GRANDPA), Algorand ...



8 – Proof-of-space and proof-of-delay

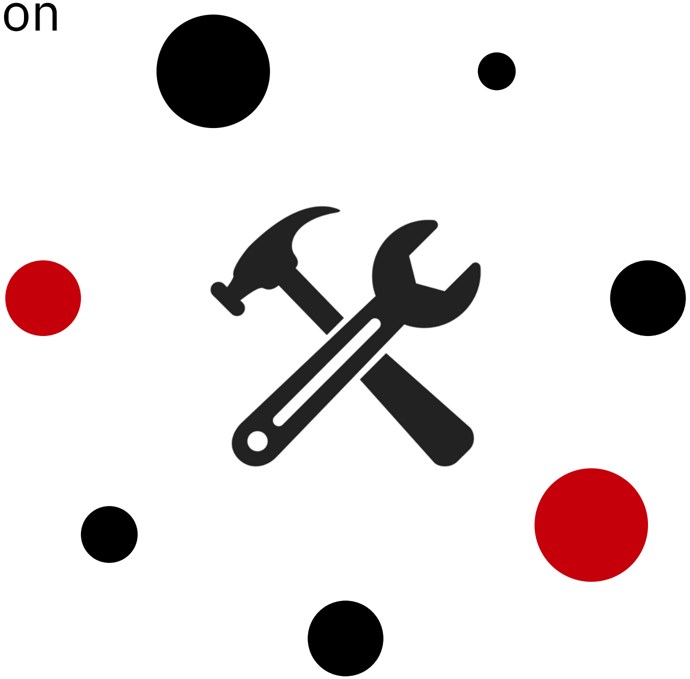
- Storage space as resource
- Cryptographic ZK proofs for storage at particular time
- Time delay to prove storage investment over time
- Filecoin, Chia, Storj ...



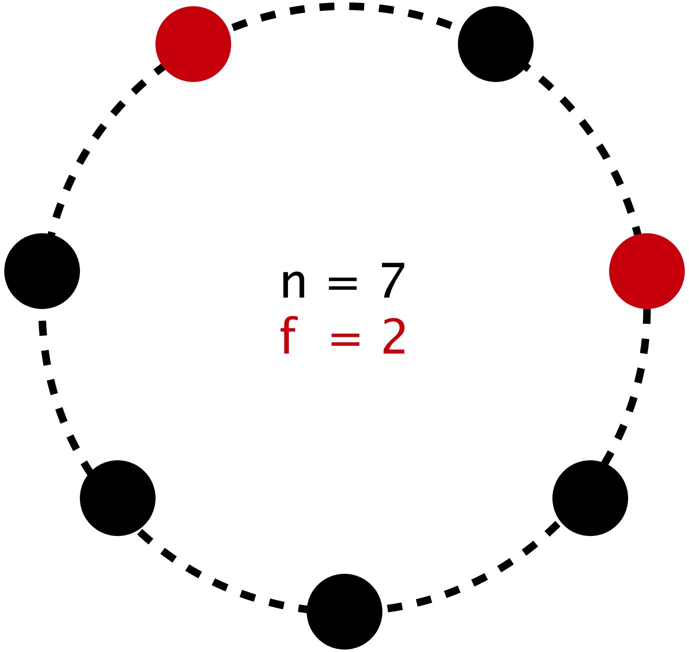
UNIVERSITÄT BERN

9 – Proof-of-work

- Demonstrate invested computation
- Nakamoto consensus
- Bitcoin ...



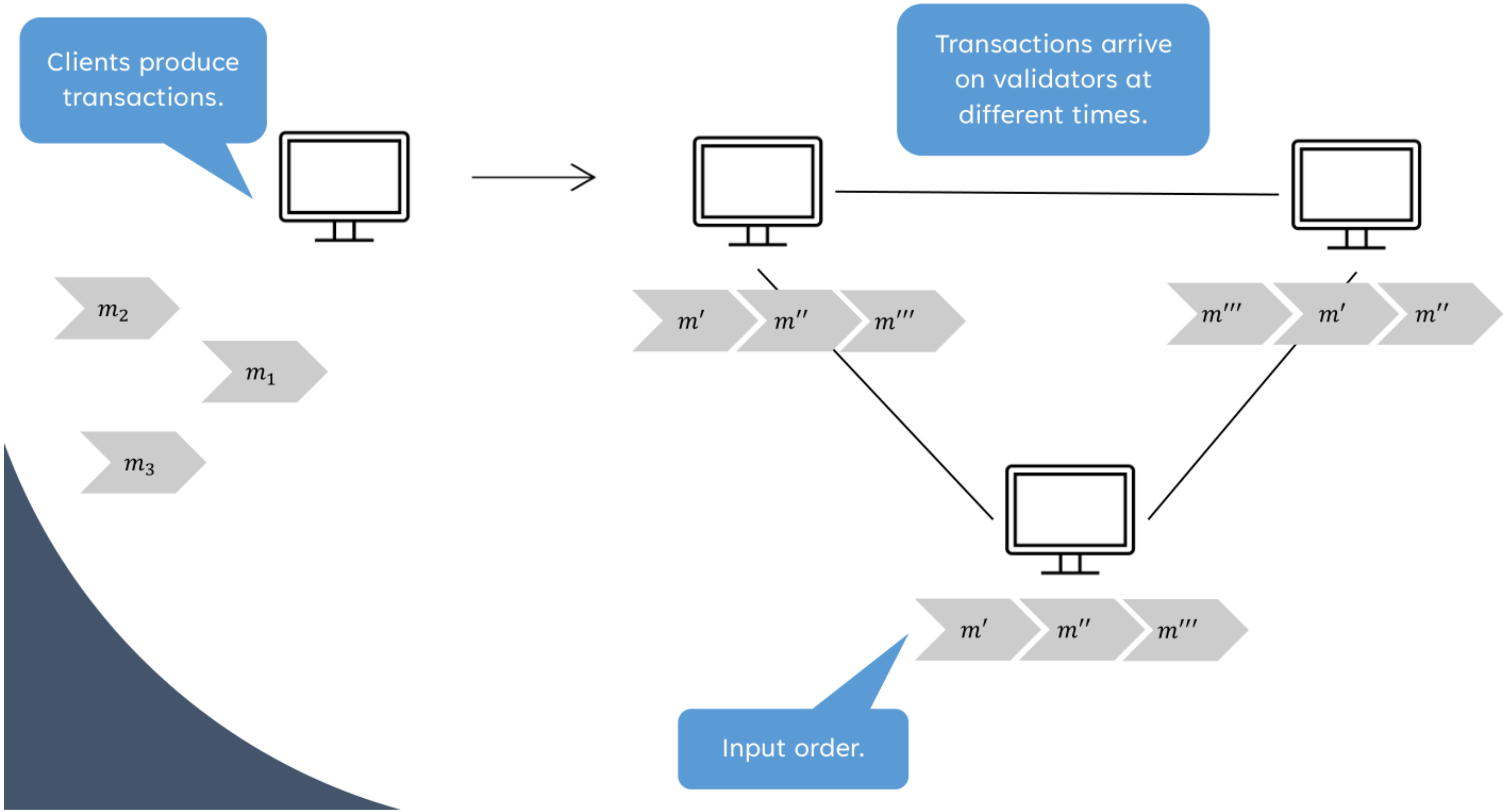
Model 1: Threshold trust



Order fairness

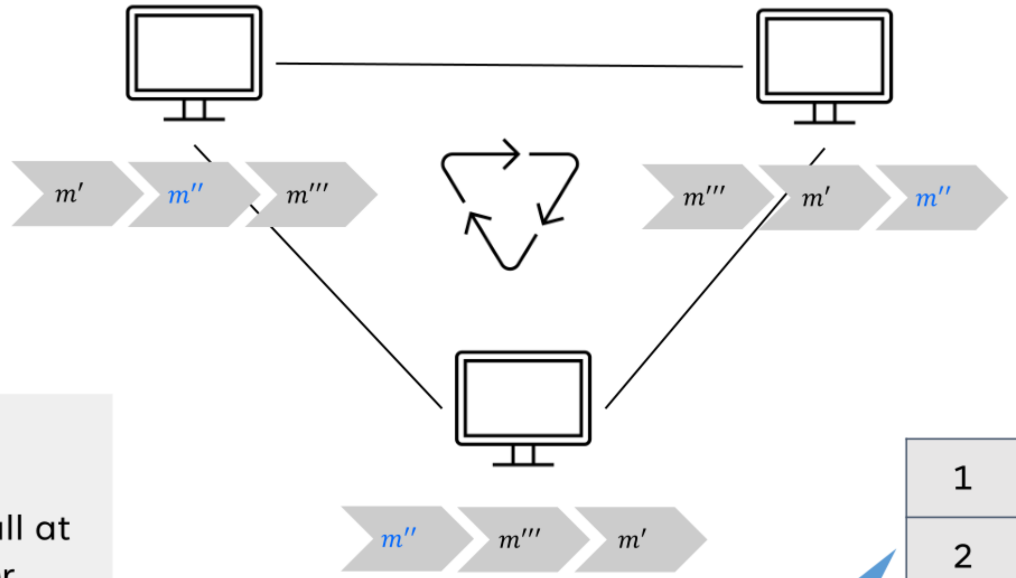
- Front-running and transaction-reordering attacks in DeFi
- Maximal extractable value (MEV)
- Validity of consensus (total-order broadcast) leaves actual order open
- Validator nodes exploit their freedom and choose a profitable order

Order fairness: Respect the receive-order



Condorcet: A fair order may not exist

UNIVERSITÄT BERN



Cycle in the order preferences!

[KZGJ20]: Fair order = deliver all at the “same time” = “block-order fairness”

1	...
2	m', m'', m'''
3	...

Output order.

Differential (block-)order fairness [CMSZ22]

UNIVERSITÄT BERN

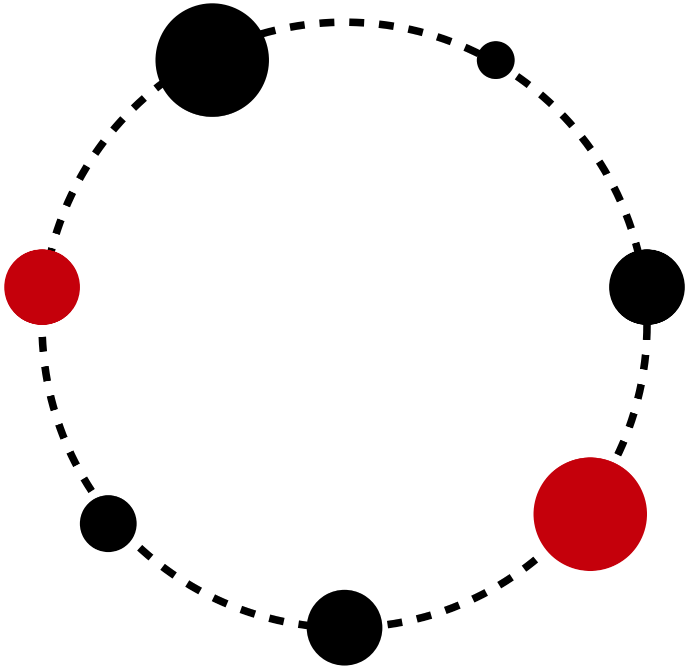
- $b(m,m')$: number of correct nodes that *receive as input* m before m'

- f out of n corrupted nodes

- **Differential order fairness**: If $b(m,m') > b(m',m) + 2f$, then no correct node *delivers* m' before m. (But protocol may *deliver* m and m' together, in same block.)

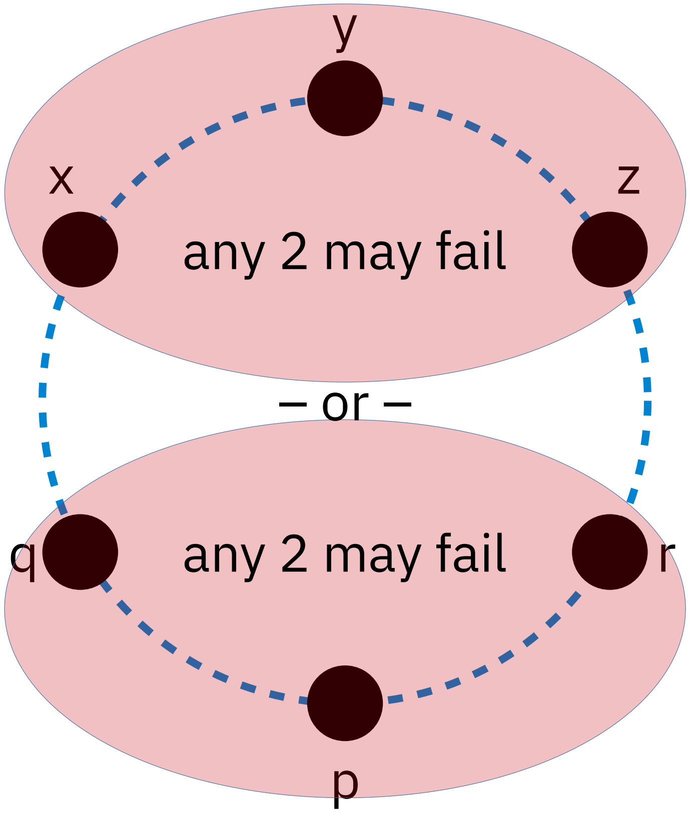
- Implemented by the quick order-fair atomic broadcast protocol, for $n > 3f$

Model 2: Generalized trust



Generalized trust – Byz. quorum systems

- Set of nodes **P** = {p_1, ..., p_n}
- Fail-prone system **F** ⊆ $2^{\mathbf{P}}$:
  – All F ∈ **F** may fail together
- Quorum system **Q** ⊆ $2^{\mathbf{P}}$, any Q ∈ **Q** is a "quorum" [MR98, HM00]
- **F** = {pq, pr, qr, xy, xz, yz}
- **Q** = {rxyz, qxyz, pxyz, pqrz, pqry, pqrx}
- Nodes are trusted **differently**
- All nodes trust **equally**



Do not trust in numbers [AC22]

- Distributed cryptography beyond the threshold model

- Theoretically well-known, practically never explored

- Example access structure (quorum set) of a validator in Stellar (SDF1)

```
{  "select": 6,
   "out-of": [
      {"select": 2, "out-of": ["Blockdaemon1", "Blockdaemon2", "Blockdaemon3"]},
      {"select": 2, "out-of": ["SDF1", "SDF2", "SDF3"]},
      {"select": 2, "out-of": ["WirexSingapore", "WirexUK", "WirexUS"]},
      {"select": 2, "out-of": ["CoinqvestFinland", "CoinqvestHongKong", "CoinqvestGermany"]},
      {"select": 2, "out-of": ["SatoshiPayUS", "SatoshiPaySG", "SatoshiPayDE"]},
      {"select": 2, "out-of": ["FranklinTempleton1", "FranklinTempleton2", "FranklinTempleton3"]},
      {"select": 3, "out-of": ["LOBSTR1", "LOBSTR2", "LOBSTR3", "LOBSTR4", "LOBSTR5"]},
      {"select": 2, "out-of": ["Hercules", "Lyra", "Boötes"]}
]}
```

Do not trust in numbers [AC22]

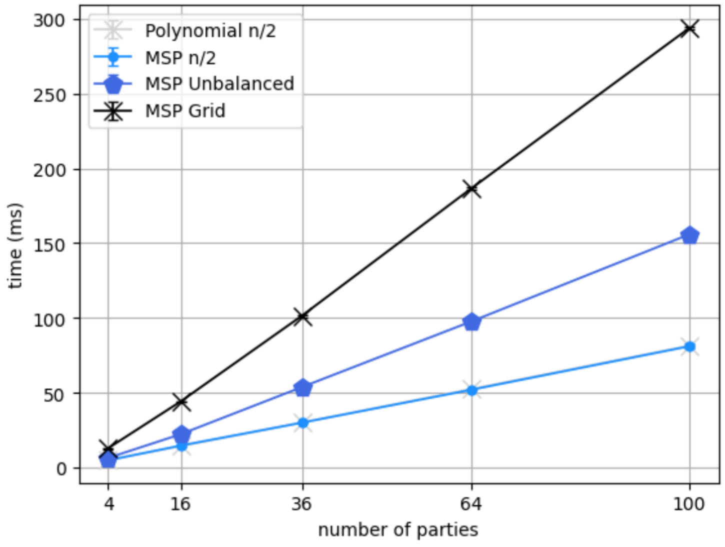
- Practical implementation of generalized cryptosystems
  - Monotone span programs (MSP)

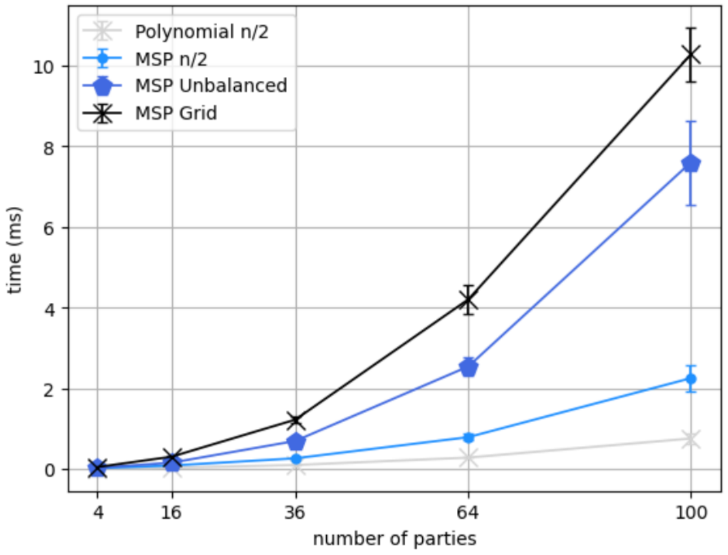
- Verifiable secret sharing (VSS)
- Common coin
- Distributed signatures

- Tools to generate MSP from a configuration file

- Benchmarks show the approach is practical

Do not trust in numbers: Verifiable Secret Sharing [AC22]

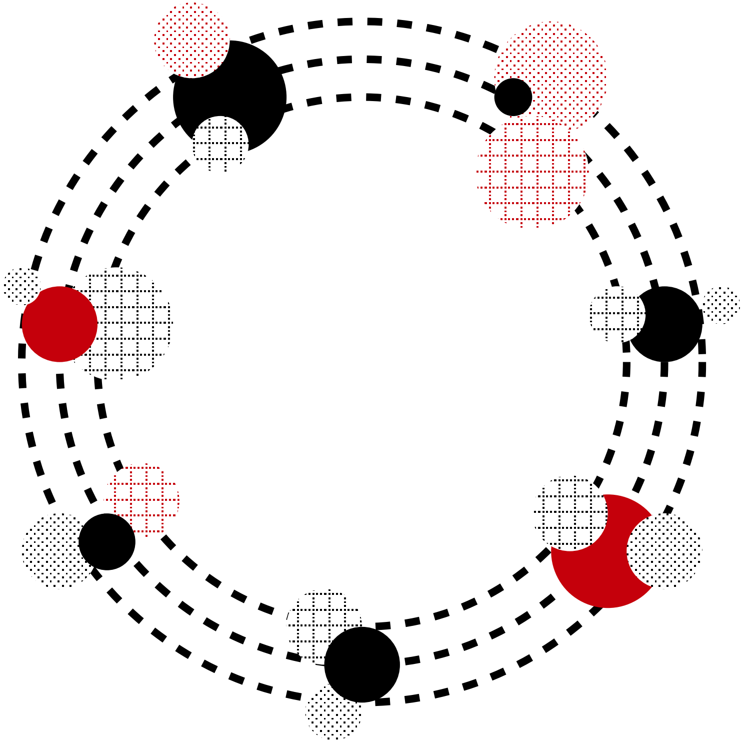




- Share and Reconstruct steps of generalized verifiable secret sharing
- Polynomial (n/2), MSP (n/2), MSP (unbalanced) and MSP (grid) structures

UNIVERSITÄT BERN

Model 3: Asymmetric trust



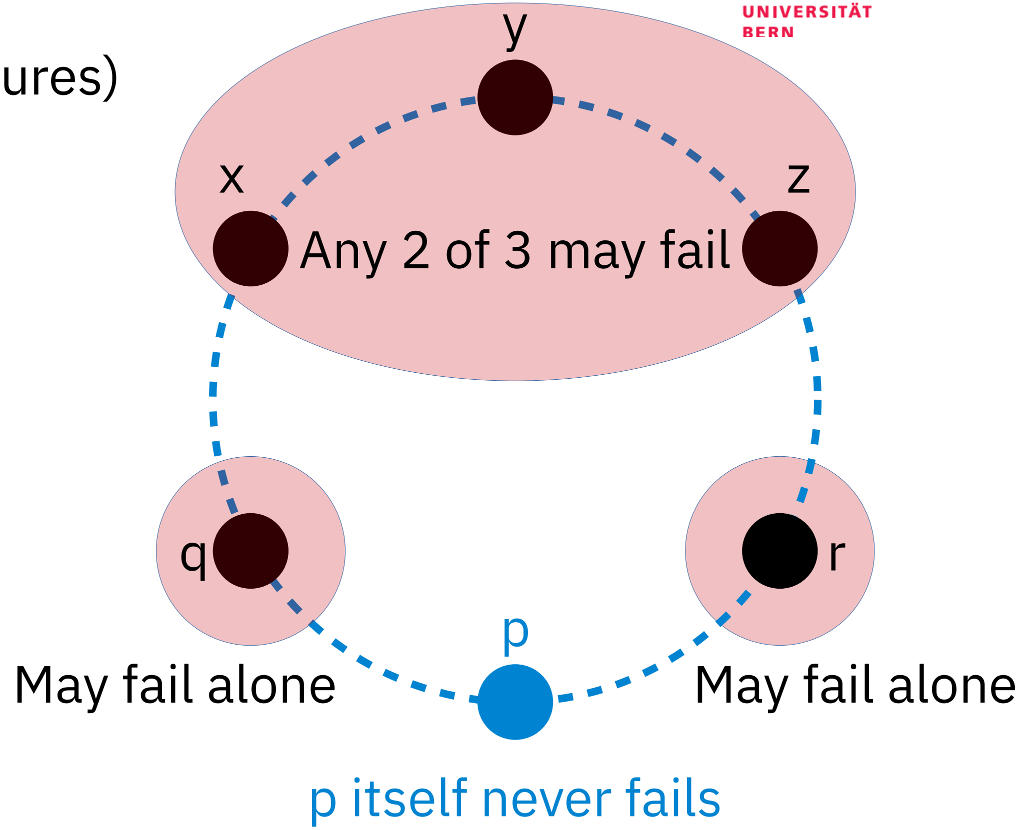
Asymmetric trust

- Subjective trust assumption of p (via failures)
  - p itself never fails
  - Neighbor nodes q and r

    May fail alone, not together with others
  - Remote nodes x, y, x

    Any 2 of these 3 may fail together
- Fail-prone system of node p

  {q, r, xy, yz, xz}



- Each one of the 6 nodes uses its own subjective trust like this

  → Asymmetric quorums

- Nodes are trusted **differently**.

Nodes trust **differently** (asymmetric).

Why asymmetric trust?

- For Romans:
  - *De gustibus non est disputandum.*
    (One cannot argue about taste.)

- For CISOs:
  - *One cannot argue about security assumptions.*

- For blockchainers:
  - *A node counts only the votes of nodes that it trusts. (Ripple, 2014)*
  - *Every node has a different idea about which other nodes are important. (Stellar, 2016)*

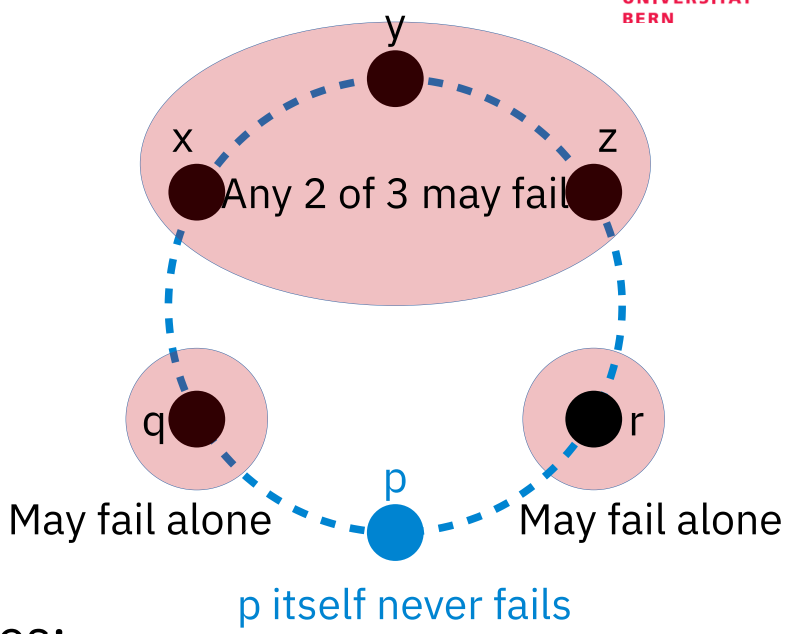
b UNIVERSITÄT BERN

Example asymmetric quorum system

- Six nodes, arranged in a ring
- Failure assumptions of node p as shown
- All others are (rotation-)symmetric to p

- Satisfies B3 property
  ⇔
  There is an asymmetric quorum system

- Each node mistrusts some 2-set of other nodes: impossible with threshold Byzantine quorums!



Execution model

- An execution defines the actually faulty nodes F

- A node p_i is one of
  - Faulty – $p_i \in F$
  - Naive p_i – $p_i \notin F$ and $F \notin$ $\mathbf{F_i}^*$
  - Wise p_i – $p_i \notin F$ and $F \in$ $\mathbf{F_i}^*$

- Safety and liveness hold only for wise nodes
  - Naive nodes may be cheated
    (cf. ordinary, symmetric model, when $f \geq n/3$: all nodes are naive!)

- Liveness depends on existence of a guild
  - A guild is a set of wise nodes that contains one quorum for each member node

SWMR regular register protocol with Byzantine processes (process p_i).

State

wts: sequence number of write operations, stored only by writer p_w
rid: identifier of read operations, used only by reader
ts, v, σ: current state stored by p_i: timestamp, value, signature

upon invocation *write*(v) **do** // only if p_i is writer p_w
$wts \leftarrow wts + 1$
$\sigma \leftarrow_\$ sign_w(\text{WRITE}\|w\|wts\|v)$
send message $[\text{WRITE}, wts, v, \sigma]$ to all $p_j \in \mathcal{P}$
wait for receiving a message $[\text{ACK}]$ from *more* than $\frac{n+f}{2}$ processes

upon invocation *read* **do** // only if p_i is reader p_r
$rid \leftarrow rid + 1$
send message $[\text{READ}, rid]$ to all $p_j \in \mathcal{P}$
wait for receiving messages $[\text{VALUE}, r_j, ts_j, v_j, \sigma_j]$ from *more* than $\frac{n+f}{2}$ processes **such that**
$r_j = rid$ **and** $verify_w(\sigma_j, \text{WRITE}\|w\|ts\|v_j)$
return *highestval*$(\{(ts_j, v_j)\})$

upon receiving a message $[\text{WRITE}, ts', v', \sigma']$ from p_w **do** // every process
if $ts' > ts$ **then**
$(ts, v, \sigma) \leftarrow (ts', v', \sigma')$
send message $[\text{ACK}]$ to p_w

upon receiving a message $[\text{READ}, r]$ from p_r **do** // every process
send message $[\text{VALUE}, r, ts, v, \sigma]$ to p_r

Asymmetric SWMR regular register protocol (process p_i).

State

wts: sequence number of write operations, stored only by writer p_w
rid: identifier of read operations, used only by reader
ts, v, σ: current state stored by p_i: timestamp, value, signature

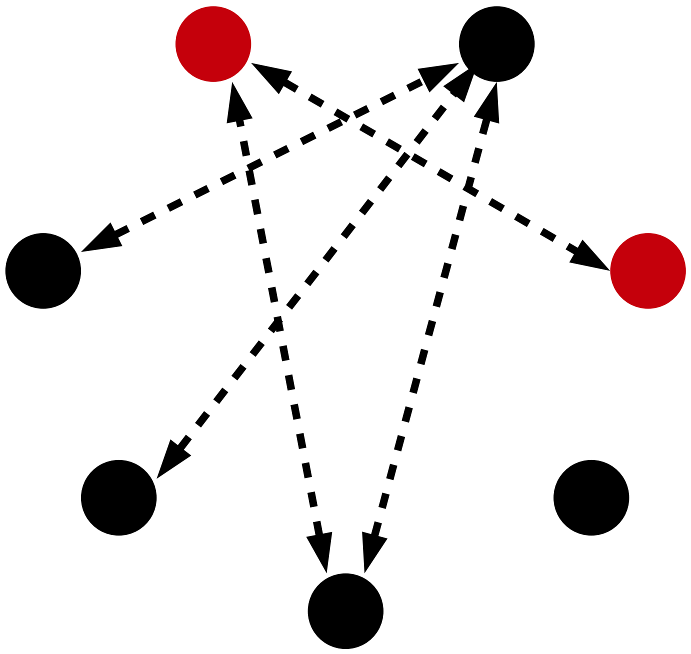
upon invocation *write*(v) **do** // only if p_i is writer p_w
$wts \leftarrow wts + 1$
$\sigma \leftarrow_{\$} sign_w(\text{WRITE}\|w\|wts\|v)$
send message $[\text{WRITE}, wts, v, \sigma]$ to all $p_j \in \mathcal{P}$
wait for receiving a message $[\text{ACK}]$ from all processes in some quorum $Q_w \in \mathcal{Q}_w$

upon invocation *read* **do** // only if p_i is reader p_r
$rid \leftarrow rid + 1$
send message $[\text{READ}, rid]$ to all $p_j \in \mathcal{P}$
wait for receiving messages $[\text{VALUE}, r_j, ts_j, v_j, \sigma_j]$ from all processes in some $Q_r \in \mathcal{Q}_r$ **such that**
$r_j = rid$ **and** $verify_w(\sigma_j, \text{WRITE}\|w\|ts\|v_j)$
return $highestval(\{(ts_j, v_j) | j \in Q_r\})$

upon receiving a message $[\text{WRITE}, ts', v', \sigma']$ from p_w **do** // every process
if $ts' > ts$ **then**
$(ts, v, \sigma) \leftarrow (ts', v', \sigma')$
send message $[\text{ACK}]$ to p_w

upon receiving a message $[\text{READ}, r]$ from p_r **do** // every process
send message $[\text{VALUE}, r, ts, v, \sigma]$ to p_r

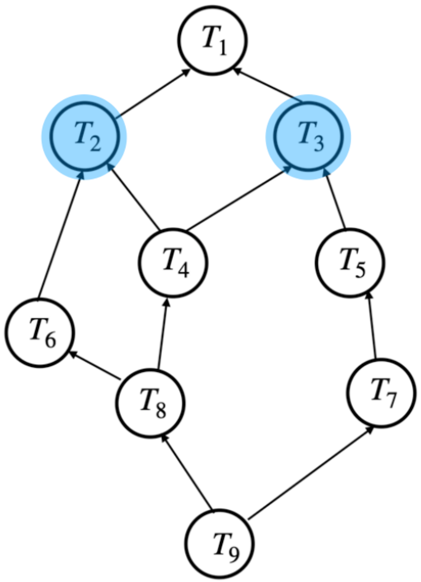
Model 4: Unstructured, probabilistic voting



Analysis of Avalanche consensus [ACT22]

UNIVERSITÄT
BERN

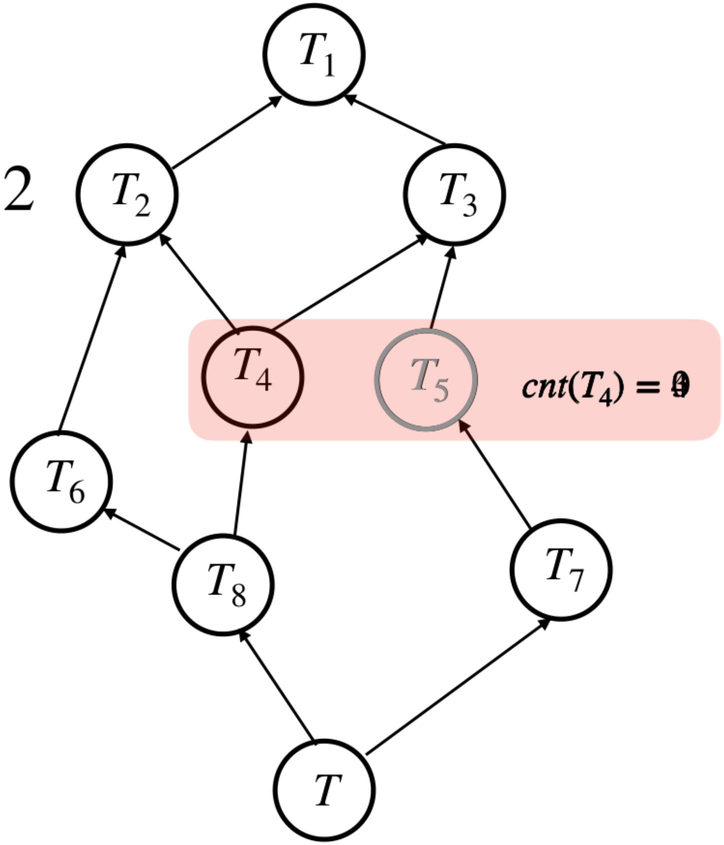
- Metastable consensus: Avalanche and the snow family of protocols
- Transactions form a DAG,
  a directed acyclic graph
- Transactions may conflict
- Nodes sample other nodes
  and ask for their opinion



Avalanche consensus

- While true do
  - Select a new transaction T
  - Query k parties with T
  - If more than α positive answers
    - Update the DAG and increment the counter for acceptance of every ancestor
  - Else
    - Reset the counter for acceptance of every ancestor to 0

$$2 \geq \alpha = 2$$

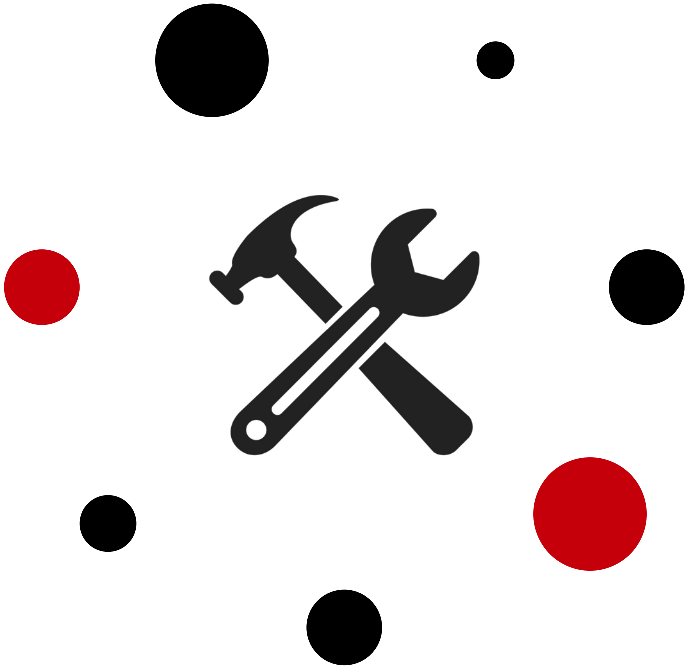


Analysis of Avalanche consensus [ACT22]

b
UNIVERSITÄT
BERN

- Detailed pseudocode of Avalanche protocol
- Independent analysis
- Illustrates a potential problem
- For other reasons, Ava Labs/Avalanche abandons the DAG protocol in March '23

Model 9: Proof-of-work



Use any resource for consensus [ACLVZ22]

- Longest-chain consensus based on an abstract resource
- Formal model of a resource allocator
- Resources: work, stake, storage ...
- Which features must a resource have to enable consensus?

Thank you!

Web – https://crypto.unibe.ch/

Blog – https://cryptobern.github.io/

Twitter – https://twitter.com/cczurich/

References

Literature

UNIVERSITÄT BERN

- Model 1: Threshold trust
  - [CMSZ22] Cachin, C., Mićić, J., Steinhauer, N. & Zanolini, L. (2022). Quick Order Fairness. In I. Eyal & J. A. Garay (Eds.), Proc. Financial Cryptography and Data Security (FC) (Vol. 13411, pp. 316–333). Springer. https://doi.org/10.1007/978-3-031-18283-9_15

Literature

UNIVERSITÄT BERN

- Model 2: Generalized trust

  - [AC20] Alpos, O., & Cachin, C. (2020). Consensus Beyond Thresholds: Generalized Byzantine Quorums Made Live. Proc. 39th Symposium on Reliable Distributed Systems (SRDS), 31–40. https://doi.org/10.1109/SRDS51746.2020.00010

  - [ACZ21] Alpos, O., Cachin, C., & Zanolini, L. (2021). How to Trust Strangers: Composition of Byzantine Quorum Systems. Proc. 40th Symposium on Reliable Distributed Systems (SRDS), 120–131. https://doi.org/10.1109/SRDS53918.2021.00021

  - [AC22] Alpos, O., & Cachin, C. (2022). Do Not Trust in Numbers: Practical Distributed Cryptography With General Trust. Cryptology ePrint Archive, Report 2022/1767. https://eprint.iacr.org/2022/1767. To appear in Proc. SSS 2023.

Literature

- Model 3: Asymmetric trust

  - [AZ21] Cachin, C., & Zanolini, L. (2021). Asymmetric Asynchronous Byzantine Consensus. Proc. ESORICS Workshops on Data Privacy Management (DPM), Cryptocurrencies and Blockchain Technology (CBT) (Vol. 13140, pp. 192–207). Springer. https://doi.org/10.1007/978-3-030-93944-1_13

  - [AZ20] Cachin, C., & Zanolini, L. (2020). From Symmetric to Asymmetric Asynchronous Byzantine Consensus. e-print, arXiv:2005.08795v3 [cs.DC]. https://arxiv.org/abs/2005.08795v3

  - [CLZ22] Cachin, C., Losa, G., & Zanolini, L. (2022). Quorum Systems in Permissionless Proc. 26th International Conference on Principles of Distributed Systems (OPODIS) (Vol. 253, pp. 17:1–17:22). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.OPODIS.2022.17

Literature

UNIVERSITÄT BERN

- Model 3: Asymmetric trust
  - [ACM21] Amores-Sesar, I., Cachin, C., & Mićić, J. (2021). Security Analysis of Ripple Consensus. Proc. 24th International Conference on Principles of Distributed Systems (OPODIS) (Vol. 184, pp. 10:1–10:16). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.OPODIS.2020.10

Literature

UNIVERSITÄT BERN

- Model 4: Unstructured, probabilistic voting
  - [ACT22] Amores-Sesar, I., Cachin, C., & Tedeschi, E. (2022). When is Spring coming? A Security Analysis of Avalanche Consensus. Proc. 26th International Conference on Principles of Distributed Systems (OPODIS) (Vol. 253, pp. 10:1–10:22). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.OPODIS.2022.10

Literature

- Models 5-7: Stake based
  - [B21] Bürk, T. (2022). Blockchain consensus protocols based on stake. Master thesis, Institute of Computer Science, University of Bern. https://crypto.unibe.ch/archive/theses/2021.msc.timo.buerk.pdf

Literature

UNIVERSITÄT BERN

- Model 9

  - [ACP21] Amores-Sesar, I., Cachin, C., & Parker, A. (2021). Generalizing Weighted Trees: A Bridge from Bitcoin to GHOST. In F. Baldimtsi & T. Roughgarden (Eds.), Proc. 3rd ACM Conference on Advances in Financial Technologies (AFT) (pp. 156–169). https://doi.org/10.1145/3479722.3480995

  - [ACLVZ22] Azouvi, S., Cachin, C., Le, D. V., Vukolic, M., & Zanolini, L. (2022). Modeling Resources in Permissionless Longest-Chain Total-Order Broadcast. In E. Hillel, R. Palmieri, & E. Rivière (Eds.), Proc. 26th International Conference on Principles of Distributed Systems (OPODIS) (Vol. 253, pp. 19:1–19:23). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.OPODIS.2022.19