Proof-of-Work-based Consensus in Expected-Constant Time

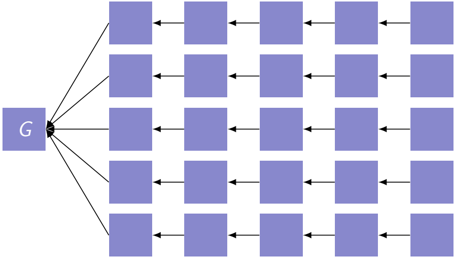
https://eprint.iacr.org/2023/1663

EUROCRYPT 2024

[1]Texas A&M University [2]University of Edinburgh [3]IOG

Juan Garay[1], Aggelos Kiayias[2,3], Yu Shen[2]

Main Results



Blockchain-based consensus in **expected-constant rounds**.

- Previously: polylog(κ) rounds.
- Implies faster transaction confirmation on distributed ledgers.

1. **Proof-of-Work-based Consensus**

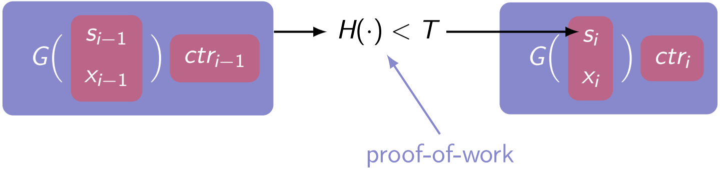
2. Consensus in Expected-Constant Time

3. Parallel Blockchains

4. Summary

Proofs of Work (aka "Crypto Puzzles")

- Moderately hard functions: Spam mitigation, denial of service protection, ...
- Most impactful application: Design of blockchain protocols such as Bitcoin

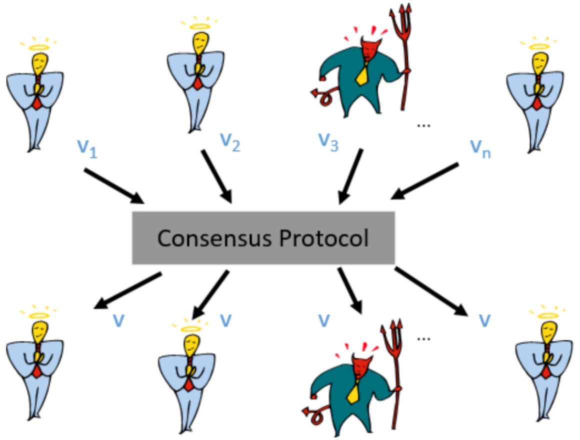


Random oracle (RO) ← $\text{Hash}(ctr_{i-1}; \text{Hash}(s_{i-1}, x_{i-1})) < T$

Consensus (aka Byzantine Agreement) [PSL80; LSP82]

n parties

t corrupted



Agreement:

All honest parties output the same value.

Validity:

If all parties start with the same value, then output that value.

Termination:

Parties eventually terminate.

On the Necessity of a PKI ("Private-State Setup")

- Consensus is impossible with $t \geq n/3$ assuming no cryptography (i.e., digital signatures) is used [PSL80; LSP82].
- The bound on no. of corruptions can be improved to $t < n/2$ using a Public Key Infrastructure (PKI) — called "private (state) setup".
- Without a PKI, consensus is impossible when $t \geq n/3$ even if using cryptography [Bor96].

[PSL80] Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. "Reaching Agreement in the Presence of Faults".
[LSP82] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. "The Byzantine Generals Problem".
[Bor96] Malte Borderding. "Levels of Authentication in Distributed Agreement".

On the Necessity of a PKI (“Private-State Setup”)

- Consensus is impossible with $t \geq n/3$ assuming no cryptography (i.e., digital signatures) is used [PSL80; LSP82].
- The bound on no. of corruptions can be improved to $t < n/2$ using a Public Key Infrastructure (PKI) — called “private (state) setup”.
- Without a PKI, consensus is impossible when $t \geq n/3$ even if using cryptography [Bor96].
- These results were established over 20 years ago...

[PSL80] Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. “Reaching Agreement in the Presence of Faults”.

[LSP82] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. “The Byzantine Generals Problem”.

[Bor96] Malte Borderding. “Levels of Authentication in Distributed Agreement”.

Nakamoto's Proposal

- "The proof-of-work chain is a solution to the Byzantine Generals Problem..."

> A number of Byzantine Generals each have a computer and want to attack the King's wi-fi by brute forcing the password, which they've learned is a certain number of characters in length. Once they stimulate the network to generate a packet, they must crack the password within a limited time to break in and erase the logs, lest they be discovered. They only have enough CPU power to crack it fast enough if a majority of them attack at the same time.
> They don't particularly care when the attack will be, just that they agree. It has been decided that anyone who feels like it will announce an attack time, which we'll call the "plan", and whatever plan is heard first will be the official plan. The problem is that the network is not instantaneous, and if two generals announce different plans at close to the same time, some may hear one first and others hear the other first.

Nakamoto's Proposal (Cont'd)

- Parties start building a blockchain inserting their input. If a party receives a longer blockchain, it switches to that one and **switches its input**. When the blockchain is long enough, the party outputs the (unique) value that it contains.

Nakamoto's Proposal (Cont'd)

- Parties start building a blockchain inserting their input. If a party receives a longer blockchain, it switches to that one and **switches its input**. When the blockchain is long enough, the party outputs the (unique) value that it contains.

- **Issue:** If adv. finds a solution first, then honest parties will extend adv.'s solution and switch to adv.'s input.

→ Protocol doesn't guarantee **validity** with overwhelming probability.

→ **Nakamoto's proposal does NOT solve consensus.**

First PoW-based Consensus Protocol [GKL15]

- Parties start building a blockchain inserting their input. If a party receives a longer blockchain, it switches to that one but **keeps the same input**. When the blockchain is long enough, the party outputs the majority value in its prefix.
  - **Agreement** from **Common Prefix**.
  - **Validity** as long as adv. controls $< 1/3$ of the parties (tight, due to **Chain Quality**).

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. **"The Bitcoin Backbone Protocol: Analysis and Applications".**

First PoW-based Consensus Protocol [GKL15]

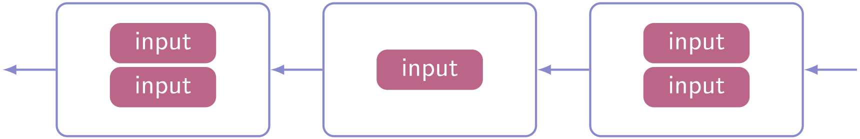
- Parties start building a blockchain inserting their input. If a party receives a longer blockchain, it switches to that one but **keeps the same input**. When the blockchain is long enough, the party outputs the majority value in its prefix.
  - **Agreement** from **Common Prefix**.
  - **Validity** as long as adv. controls $< 1/3$ of the parties (tight, due to **Chain Quality**).

- $1/3$ is **suboptimal**.
  - **Main obstacle:** The blockchain does not provide sufficient **chain quality**.
- $1/2$ can be achieved, using a more elaborate protocol — 2×1 **PoWs**.

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. **"The Bitcoin Backbone Protocol: Analysis and Applications".**

1/2 Consensus Protocol

- Parties mine PoWs for each **block** — as in standard Bitcoin backbone protocol
- Parties mine PoWs for each **input in** $\{0,1\}$ (with nonce); they keep transmitting **"PoW-ed" inputs** until they are accepted.



- After the blockchain grows sufficiently, they chop off the last $k = \mathrm{polylog}(\kappa)$ blocks and return the majority among unique inputs in the common prefix.

1/2 Consensus Protocol (Cont'd)

- **Beware!**
  Given that PoWs would be used for two different tasks, how do we prevent the adversary from shifting his computer power from to the other?

1/2 Consensus Protocol (Cont'd)

- **Beware!**
  Given that PoWs would be used for two different tasks, how do we prevent the adversary from shifting his computer power from to the other?

- **...with 2×1 PoWs!**

2x1 PoWs: Composition of PoW-based Protocols

Naïve double PoW (Not secure!)

$h \leftarrow G(x, s)$

if $H(h, ctr) < T$ then ...

$h' \leftarrow G(x', s')$

if $H(h', ctr') < T'$ then ...

Given $((x, s), ctr)$

Verify $H(G(x, s), ctr) < T$

Given $((x', s'), ctr')$

Verify $H(G(x', s'), ctr') < T'$

2×1 PoW

$h \leftarrow G(x, s)$

$h' \leftarrow G(x', s')$

$w \leftarrow H(h, h', ctr)$

if $w < T$ then ...

if $[w]^{R} < T'$ then ...

Given $((x, s), (*, *), ctr)$

Verify $H(G(x, s), G(*, *), ctr) < T$

Given $((*, *), (x', s'), ctr')$

Verify $H(G(*, *), G(x', s'), ctr') < T'$

Round Complexity of Byzantine Agreement

- Deterministic BA
  - Requires $(t+1)$ rounds. [FL82; DS83]
  - Composes nicely.
- Randomization can help. [Rab83]
  - BA from OCC (oblivious common coin) tolerating $t < n/3$ corruptions. [FM88]
  - BA from OLE (oblivious leader election) tolerating $t < n/2$ corruptions. [KK06]

[FL82] Michael J. Fischer and Nancy A. Lynch. "A Lower Bound for the Time to Assure Interactive Consistency".

[DS83] Danny Dolev and H. Raymond Strong. "Authenticated Algorithms for Byzantine Agreement".

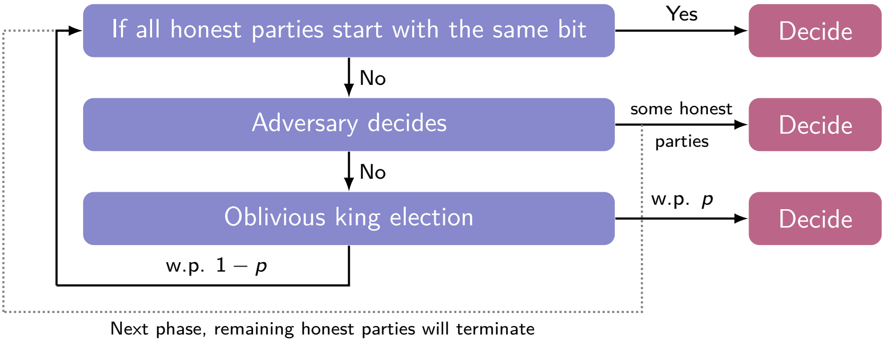
[Rab83] Michael O. Rabin. "Randomized Byzantine Generals".

[FM88] Paul Feldman and Silvio Micali. "Optimal Algorithms for Byzantine Agreement".

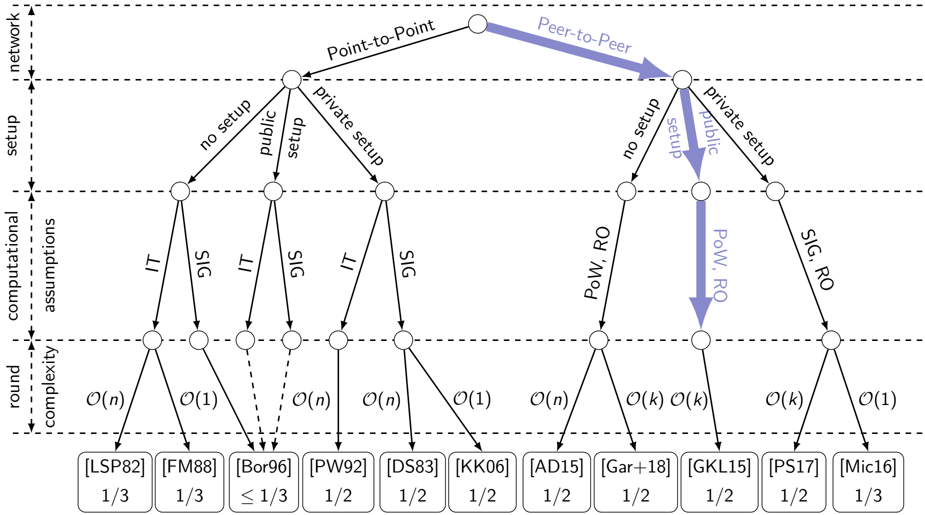
[KK06] Jonathan Katz and Chiu-Yuen Koo. "On Expected Constant-Round Protocols for Byzantine Agreement".

King Consensus [BGP89; FG03]

- Proceeds in phases until termination.
- In each phase each party has an input bit.



A Consensus Taxonomy [GK20]



Round Complexity of PoW-based Consensus

Protocol	Setup & assumptions	Round complexity
[AD15]	RO + SIG	$\mathcal{O}(n)$
[GKL15]	CRS + RO	$\mathcal{O}(\text{polylog}\kappa)$
[Gar+18]	RO	$\mathcal{O}(\text{polylog}\kappa)$
[Das+22]	RO + SIG + VDF	Expected $\mathcal{O}(1)$
[GKS24]	CRS + RO	Expected $\mathcal{O}(1)$

[AD15] Marcin Andrychowicz and Stefan Dziembowski. "PoW-Based Distributed Cryptography with No Trusted Setup".

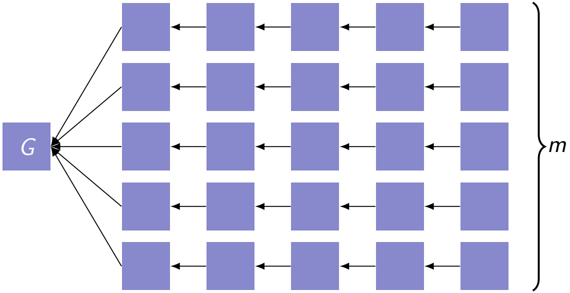
[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications".

[Gar+18] Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, and Giorgos Panagiotakos. "Bootstrapping the Blockchain, with Applications to Consensus and Fast PKI Setup".

[Das+22] Poulami Das, Lisa Eckey, Sebastian Faust, Julian Loss, and Monosij Maitra. "Round Efficient Byzantine Agreement from VDFs".

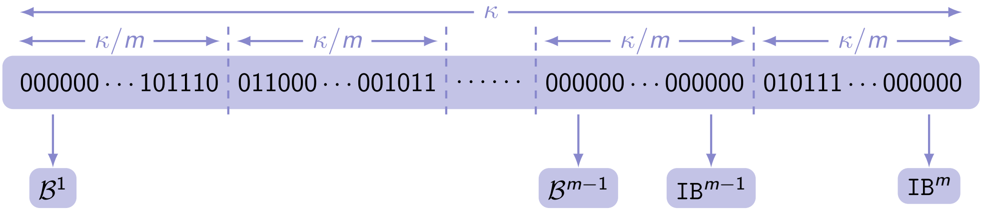
Parallel Blockchains

- **Basic Idea:** Extending 2×1 PoW to $m\times 1$ **PoW**.
- Fully independent when $m = \Theta(\text{polylog}\kappa)$.



Parallel Blockchains (Cont'd)

- **Basic Idea:** Extending 2×1 PoW to m×1 **PoW**.
- Fully independent when $m = \Theta(\text{polylog}\kappa)$.
- We can run PoW BAs in parallel.
  - 2×1 PoW (block + transaction) in each instance.



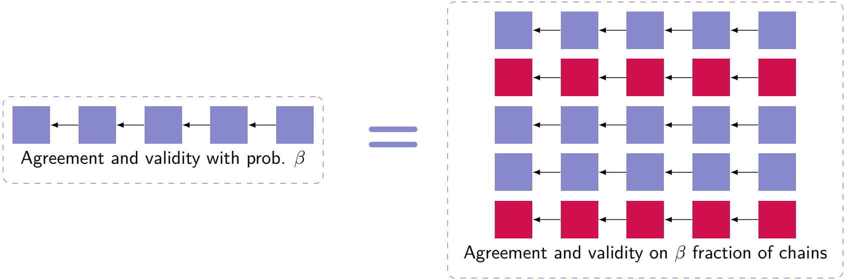
Phase-based Parallel Chains

- Recall honest-majority PoW consensus [GKL15]:
  - Agreement and validity with **overwhelming** prob. after polylog rounds.
  - Agreement and validity with **constant** prob. after **constant** rounds.

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications".

Phase-based Parallel Chains

- Recall honest-majority PoW consensus [GKL15]:
  - Agreement and validity with **overwhelming** prob. after polylog rounds.
  - Agreement and validity with **constant** prob. after **constant** rounds.
- With sufficently many parallel chains:



[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications".

Phase-based Parallel Chains (Cont'd)

- A phase consists of constant ρ rounds.
- In each phase, a β fraction of chains achieves agreement and validity **obliviously**.
- ✓ Good for **validity** if $\beta > 1/2$.

Phase-based Parallel Chains (Cont'd)

- A phase consists of constant ρ rounds.
- In each phase, a β fraction of chains achieves agreement and validity **obliviously**.

✓ Good for **validity** if $\beta > 1/2$.

✗ Not "good enough" for **agreement** (even if $\beta < 1$ is an **arbitrary constant**)

  - Half 1s and half 0s $\Longrightarrow$ output dominated by a chain controlled by the adversary.

Phase-based Parallel Chains (Cont'd)

- A phase consists of constant ρ rounds.
- In each phase, a β fraction of chains achieves agreement and validity **obliviously**.

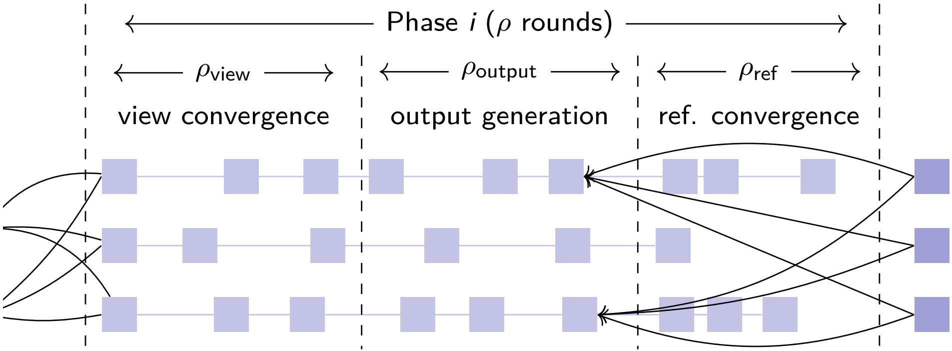
✓ Good for **validity** if $\beta > 1/2$.

✗ Not "good enough" for **agreement** (even if $\beta < 1$ is an **arbitrary constant**)

- Half 1s and half 0s $\implies$ output dominated by a chain controlled by the adversary.

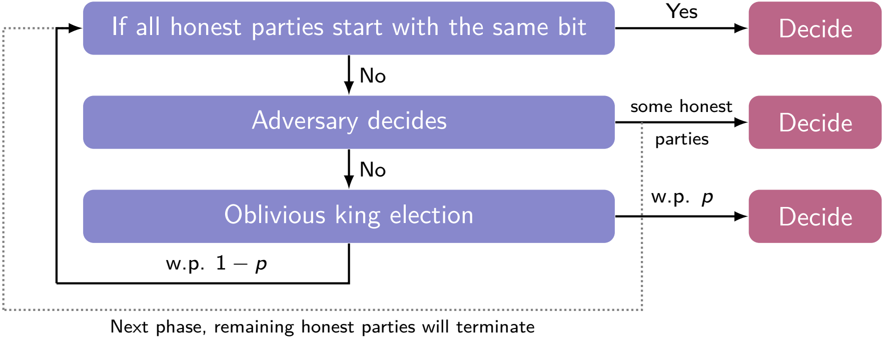
→ Use phases to emulate rounds in classical protocols!

Phase-based Parallel Chains (Cont'd)



King Consensus [BGP89; FG03]

- Proceeds in phases until termination.
- In each phase each party has an input bit.

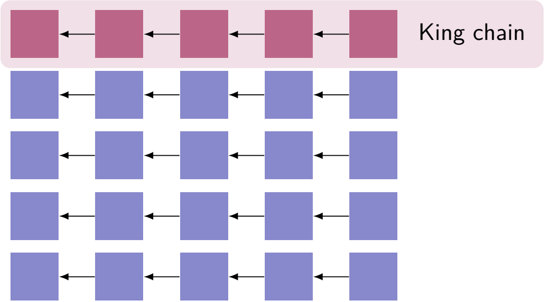


Chain-King Consensus

- Oblivious leader election (OLE) using only RO?

Chain-King Consensus

- Oblivious leader election (OLE) using only RO?
- A simple construction: Fix the **1st chain** as the "King Chain".
- With parallel chains, adversary power is "**diluted**" so that he cannot always win on a specific chain.



Chain-King Consensus (Cont'd)

- Oblivious leader election (OLE) using only RO?
- A simple construction: Fix the **1st chain** as the "King Chain".
- With parallel chains, adversary power is "**diluted**" so that he cannot always win on a specific chain.

Chain-King Consensus

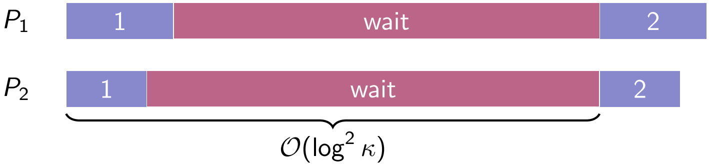
Phase-based parallel chains	Randomized king consensus	Fix 1st chain as king chain

Fast Sequential Composition

- After an invocation of chain-king consensus, parties might terminate **non-simultaneously**.
  - Security holds only when parties start **at the same time**.
- Parallel composition: how to securely start the second and later invocations?

Fast Sequential Composition

- After an invocation of chain-king consensus, parties might terminate **non-simultaneously**.
  - Security holds only when parties start **at the same time**.
- Parallel composition: how to securely start the second and later invocations?
- Naïve solution:
  - Wait for re-synchronization: running the protocol for **polylog** rounds $\Longrightarrow$ all parties terminate with overwhelming probability.

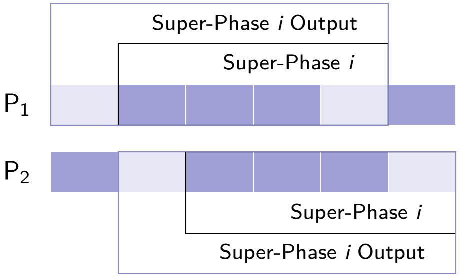


Fast Sequential Composition (Cont'd)

- Our approach: Bracha termination + **super-phase expansion**.
- Bracha termination: reduce any c-slack to $c = 1$.
- Super-phase expansion
  - Expand a phase to a **super-phase** of $(3c + 1)$ phases: $(2c + 1)$ working-phases plus c dummy phases.
- Output of a super-phase
  - Look at $(4c + 1)$ phases in local view (starting from c phases ahead of the current super-phase), the output what the $(c + 1)$**-th non-**$\perp$ phase outputs.
  - **Intuition:** Honest parties adopt output from the same phase when listening to the king chain.

Fast Sequential Composition (Cont'd)

- Output of a super-phase
  - Look at $(4c+1)$ phases in local view (starting from c phases ahead of the current super-phase), the output what the **$(c+1)$-th non-$\perp$** phase outputs.
  - **Intuition:** Honest parties adopt output from the same phase when listening to the king chain.



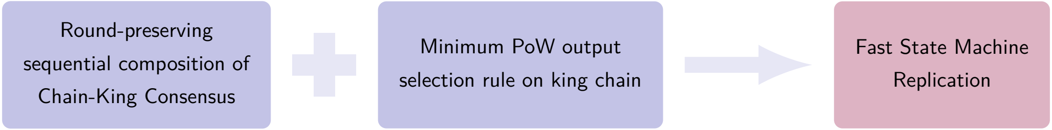
Example: Super-phase output extraction when $c = 1$

Fast State Machine Replication (Ledger Consensus)

- Decide output of king chain using input-block with minimum PoW (smallest hash).
  - With **constant prob.**, an invocation of chain-king consensus outputs a batch of transactions proposed by **honest parties**.

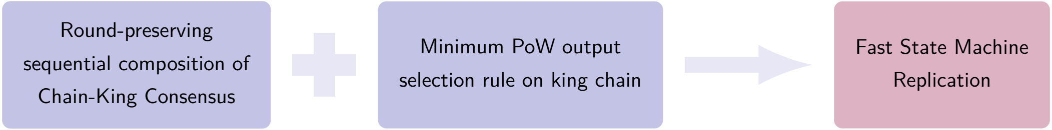
Fast State Machine Replication (Ledger Consensus)

- Decide output of king chain using input-block with minimum PoW (smallest hash).
  - With **constant prob.**, an invocation of chain-king consensus outputs a batch of transactions proposed by **honest parties**.



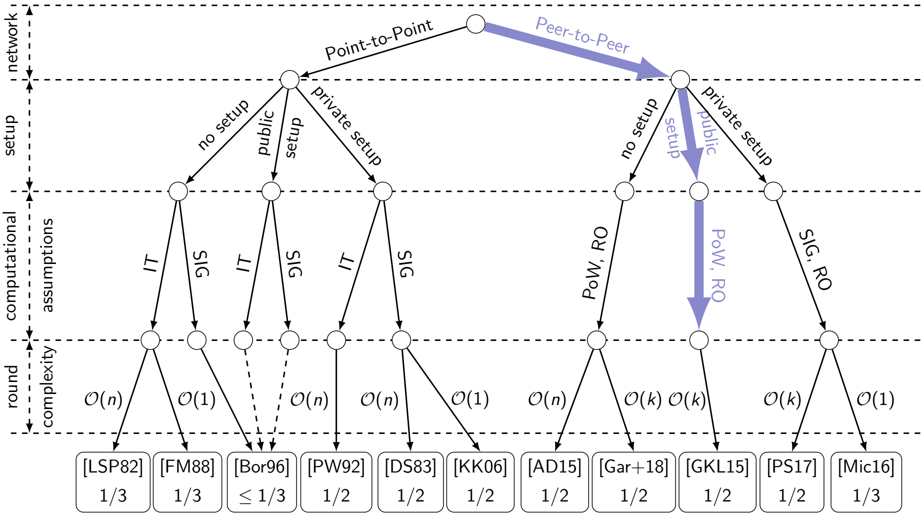
Fast State Machine Replication (Ledger Consensus)

- Decide output of king chain using input-block with minimum PoW (smallest hash).
  - With **constant prob.**, an invocation of chain-king consensus outputs a batch of transactions proposed by **honest parties**.

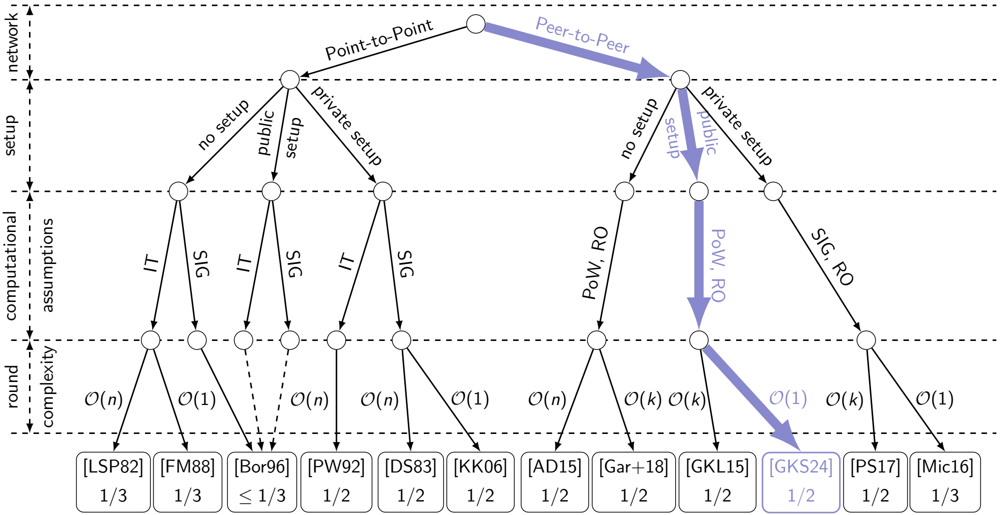


- In the same setting as Bitcoin, **all** transactions can be confirmed in **expected-constant** time.
  - In contrast, previous works only achieve constant settlement time for **non-conflicting** trasnactions, but degrade to **polylog time** with **conflicting** ones.

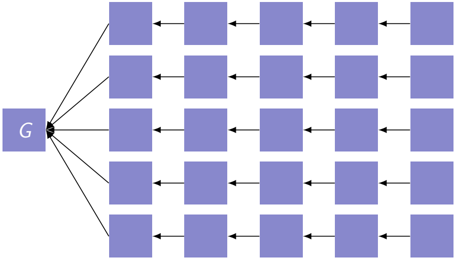
Summary & Future Directions



Summary & Future Directions



Summary & Future Directions



- Blockchain-based consensus in **expected-constant rounds**.
- Coming soon...
  - A new difficulty adjustment design, allowing for **dynamic participation**.
  - Optimal clock synchronization, improving the clock's skew from polylog(κ) to **constant**.

Thank You

Thank You!

https://eprint.iacr.org/2023/1663

References

[AD15] Marcin Andrychowicz and Stefan Dziembowski. "PoW-Based Distributed Cryptography with No Trusted Setup". In: **Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II**. Ed. by Rosario Gennaro and Matthew Robshaw. Vol. 9216. Lecture Notes in Computer Science. Springer, 2015, pp. 379–399.

[BGP89] Piotr Berman, Juan A. Garay, and Kenneth J. Perry. "Towards Optimal Distributed Consensus (Extended Abstract)". In: **30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989**. IEEE Computer Society, 1989, pp. 410–415.

[Bor96] Malte Borderding. "Levels of Authentication in Distributed Agreement". In: **Distributed Algorithms, 10th International Workshop, WDAG '96, Bologna, Italy, October 9-11, 1996, Proceedings**. Ed. by Özalp Babaoglu and Keith Marzullo. Vol. 1151. Lecture Notes in Computer Science. Springer, 1996, pp. 40–55.

[Das+22] Poulami Das, Lisa Eckey, Sebastian Faust, Julian Loss, and Monosij Maitra. "Round Efficient Byzantine Agreement from VDFs". In: **IACR Cryptol. ePrint Arch.** (2022), p. 823.

[DS83] Danny Dolev and H. Raymond Strong. "Authenticated Algorithms for Byzantine Agreement". In: **SIAM J. Comput.** 12.4 (1983), pp. 656–666.

References

[FM88] Paul Feldman and Silvio Micali. "Optimal Algorithms for Byzantine Agreement". In: **Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA**. Ed. by Janos Simon. ACM, 1988, pp. 148–161.

[FL82] Michael J. Fischer and Nancy A. Lynch. "A Lower Bound for the Time to Assure Interactive Consistency". In: **Inf. Process. Lett.** 14.4 (1982), pp. 183–186.

[FG03] Matthias Fitzi and Juan A. Garay. "Efficient player-optimal protocols for strong and differential consensus". In: **Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing, PODC 2003, Boston, Massachusetts, USA, July 13-16, 2003**. Ed. by Elizabeth Borowsky and Sergio Rajsbaum. ACM, 2003, pp. 211–220.

[GK20] Juan A. Garay and Aggelos Kiayias. "SoK: A Consensus Taxonomy in the Blockchain Era". In: **Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24-28, 2020, Proceedings**. Ed. by Stanislaw Jarecki. Vol. 12006. Lecture Notes in Computer Science. Springer, 2020, pp. 284–318.

References

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications". In: **Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II**. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. Lecture Notes in Computer Science. Springer, 2015, pp. 281–310.

[Gar+18] Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, and Giorgos Panagiotakos. "Bootstrapping the Blockchain, with Applications to Consensus and Fast PKI Setup". In: **Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II**. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10770. Lecture Notes in Computer Science. Springer, 2018, pp. 465–495.

[GKS24] Juan A. Garay, Aggelos Kiayias, and Yu Shen. "Proof-of-Work-Based Consensus in Expected-Constant Time". In: **Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part III**. Ed. by Marc Joye and Gregor Leander. Vol. 14653. Lecture Notes in Computer Science. Springer, 2024, pp. 96–125.

References

[KK06] Jonathan Katz and Chiu-Yuen Koo. "On Expected Constant-Round Protocols for Byzantine Agreement". In: **Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings**. Ed. by Cynthia Dwork. Vol. 4117. Lecture Notes in Computer Science. Springer, 2006, pp. 445–462.

[LSP82] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. "The Byzantine Generals Problem". In: **ACM Trans. Program. Lang. Syst.** 4.3 (1982), pp. 382–401.

[Mic16] Silvio Micali. "ALGORAND: The Efficient and Democratic Ledger". In: **CoRR** abs/1607.01341 (2016).

[PS17] Rafael Pass and Elaine Shi. "The Sleepy Model of Consensus". In: **Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II**. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10625. Lecture Notes in Computer Science. Springer, 2017, pp. 380–409.

[PSL80] Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. "Reaching Agreement in the Presence of Faults". In: **J. ACM** 27.2 (1980), pp. 228–234.

References

[PW92] Birgit Pfitzmann and Michael Waidner. "Unconditional Byzantine Agreement for any Number of Faulty Processors". In: **STACS 92, 9th Annual Symposium on Theoretical Aspects of Computer Science, Cachan, France, February 13-15, 1992, Proceedings**. Ed. by Alain Finkel and Matthias Jantzen. Vol. 577. Lecture Notes in Computer Science. Springer, 1992, pp. 339–350.

[Rab83] Michael O. Rabin. "Randomized Byzantine Generals". In: **24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983**. IEEE Computer Society, 1983, pp. 403–409.