OptiMine: Optimistically Sequenced Merged Mining for Rollups and Sidechains

The BOB Collective*
research@gobob.xyz

Abstract. We coin the hybrid consensus technique "Optimistically Sequenced Merged Mining". The technique enables any sidechain or rollup connected to an L1 chain to receive Proof-of-Work security from Bitcoin Miners. Rollups and sidechains retain fast block production and reduce the trust in centralized Sequencers and block-producing nodes.

1 Introduction

Scaling Bitcoin has remained a challenge for almost 15 years since its inception. With the introduction of overlay protocols, including Ordinals [13] and BRC20 [4], Bitcoin block space has become sparse, and transaction fees are rising. While alternative L1 chains like Ethereum seek scaling via a rollup-centric roadmap [2,5], Bitcoin sidechains [1] and rollups secured by its consensus remain elusive.

The critical blocker for Bitcoin-native rollups is its lack of programmability. To date, it is impractical in some and impossible in most cases to verify the state transition functions of another rollup or sidechain within Bitcoin. Eventually, Bitcoin may adopt OP codes or discover techniques to verify state transition functions of sidechains or rollups. We see promise in adopting covenants [11] [1], innovations happening in BitVM [10], and possibly integration of ZK proofs as part of Bitcoin core in the far future. However, these are not ready today.

Merged mining. An existing technique to receive Bitcoin security is through merged mining [6,7,9]. In "vanilla" merged mining, Miners submit Proof-of-Work (PoW) to two or more chains simultaneously. Thereby, an auxiliary chain inherits (some of) the security of the parent chain (e.g., Bitcoin). In the early example of Namecoin, Bitcoin Miners would submit Bitcoin blocks and block candidates with sufficient PoW to the Namecoin chain, where the coinbase transaction includes a reference to the to-be-mined Namecoin block hash. On receiving a valid auxiliary PoW, block production in Namecoin continues. RSK (Rootstock) is currently the most widely adopted merged mined chain. Each block in RSK receives PoW and block production depends on the previous merged mined block.

* Collaborative research with 349b21d947ac90d5c2d165fbde48efcee5b834597dc00609cccc2e756233481e

[1] See `https://bitcoinops.org/en/topics/covenants/` for an overview of several proposals.

A limitation of merged mining is the slow block production rate compared to Proof-of-Stake networks and L2 solutions. For example, RSK achieves a 30-second block time on average, while Optimism [2] produces a block every two seconds [3]. This limitation is due to the non-deterministic nature of Proof-of-Work, as well as the increased chance of forks due to network propagation delays when two or more blocks are found for the same high within a short period of time.

Optimistically Sequenced Merged Mining. We extend merged mining to separate block production from Proof-of-Work finalization, following the hybrid consensus model [12]. In a nutshell, blocks are produced optimistically while PoW eventually finalizes batches of blocks. Sequencers (for rollups) and consensus nodes (for sidechains) create signed blocks optimistically by collecting transactions in the network and assembling blocks. Block finalization is achieved by Bitcoin Miners running the auxiliary chain (rollup or sidechain) full node to verify the correctness of the signed blocks. Miners periodically submit PoW solutions according to the auxiliary chain difficulty. Miners finalize multiple signed blocks at once by including the signed blocks' combined hashes into the PoW solutions produced by mining, i.e., so called "mined blocks". If no valid PoW mined blocks are generated for a pre-defined timeout period, the rollup/sidechain considers this a consensus failure and block production is halted - until a mined block is found.
The protocol, termed OptiMine, achieves the following properties:

- **Bitcoin PoW security**. By running full nodes, Miners check that the auxiliary chains block producers (e.g., Sequencers, authorities, ...) have correctly created blocks. By submitting PoW that meets the mining difficulty target of the auxiliary chain (rollup or sidechain), Miners provide finality to the chain and ensure that block production can continue optimistically.
- **Fast block times**. Optimistically sequenced merged mining can achieve fast block times, e.g., two-second block times (as in the OP Stack) or sub-second block times (as in Arbitrum or Solana). Such fast block times are not economical in vanilla merged mining due to the high number of uncles produced at the low-difficulty target.
- **Recovery from Sequencer failure**. In case of a PoW failure, the auxiliary chain can fall back to the slower vanilla merged mining, whereby every block requires PoW as a recovery mechanism.
- **Bitcoin compatibility**. Optimistically sequenced merged mining is an opt-in protocol for Miners that does not require modifications to the Bitcoin software. It can be used today.

Optimistically sequenced merged mining is applicable to both rollups and sidechains. Rollups on other L1s, e.g., Ethereum, can receive Bitcoin security through this hybrid consensus by making the presence of a valid PoW part of the valid state transition function. In the case of zk-rollups, settlement of the Ethereum

[2] A rollup on Ethereum.

zk-rollup depends on the validity of a PoW submitted to the rollup within the timeout period. For optimistic rollups, a lack of recent PoW would indicate a potential Sequencer failure and halt the rollup (extensions allow for more resilient handling of failures). The settlement of the rollup on Ethereum (or other L1) inherits Bitcoin's PoW security. For sidechains, bootstrapping consensus can be improved by inheriting Bitcoin's PoW. If a sidechain uses a Proof-of-Authority (PoA) mechanism, trust in the PoA parties can be reduced by ensuring Miners validate the sidechain. In Proof-of-Stake (PoS), bootstrapping is often tricky due to the low economic security of a newly launched token or the initial low adoption of staked nodes. Here, PoW can add a layer of safety.

2 System Model and Assumptions

We assume two blockchains, parent chain X and auxiliary chain Y:

- **Parent chain X**. The parent chain employs PoW as part of its consensus protocol. For our purposes, the parent chain is Bitcoin.
- **Auxiliary chain Y** (i.e., the rollup or sidechain). The auxiliary chain receives finality from the parent chain's Miners that submit an auxiliary PoW (AuxPoW). For simplicity, we describe a single auxiliary chain below, but we note that our scheme can be applied to multiple auxiliary chains.

In our scheme, we define the following actors:

- **Sequencers**. We refer to Sequencers as the non-trusted entities that compile the existing state, incoming transactions, and external data of the auxiliary chain Y into new blocks progressing the underlying transaction ledger. Sequencers sign the produced blocks.
- **Miners**. In our context, Miners participate in the merged mining scheme. Miners run full nodes for chains X and Y and perform PoW for chain X and AuxPoW for chain Y.

We further informally define:

- **Block production**. Sequencers are entities tasked with collecting transactions and applying the state transaction functions that result in a new state. They produce signed blocks to communicate the updated state to the rest of the network and progress the network.
- **Block finalization**. Chain Y has a mining difficulty and target time. On submission of a block header of chain X to chain Y that meets the difficulty criteria, i.e., the leading 0 in the block hash of chain X, and a reference to the state of chain Y, chain Y is considered finalized.
- **Signed blocks**. Signed blocks are proposed by Sequencers.
- **Mined blocks**. Mined blocks are blocks of chain X that contain a reference to signed blocks of chain Y and meet the difficulty criteria of chain Y. A mined block finalizes a set of signed blocks.

- **Checkpoint**. A checkpoint refers to a set of signed blocks. The merged mining checkpoint can be implemented in various ways. The simplest approach is to use the hash of the Nth block, i.e., the tip of the sidechain. To achieve better provenance of the sidechain execution on Bitcoin, the entire list of N sidechain blocks can be submitted to Miners. Since including N hashes in a Bitcoin block incurs high storage costs, a more efficient approach is to use a hash, e.g., submit the root hash of a Merkle tree that stores the hashes of the N to-be-finalized signed blocks as leaves or a simple hash of a vector of hashes.
- **Block intervals**. The merged mining interval is determined by the PoW difficulty target of the auxiliary chain, i.e., Miners continuously run the mining software and submit a mined block whenever a valid PoW solution that matches the auxiliary chain's difficulty target is found.
- **Mining timeout**. Chain Y defines an upper bound of the number of non-finalized signed blocks. Sequencers stop producing blocks if after the last valid AuxPoW submission plus the timeout, no valid AuxPoW is submitted. Sequencers await the submission of a valid AuxPoW past the mining timeout to continue block production.

We make no assumptions about the auxiliary chain's block production mechanism. The auxiliary chain may (1) employ any block production mechanism, including zk and optimistic rollups on another chain that is not X, or (2) be a chain with an independent set of block producers.

3 Optimistically Sequenced Merged Mining

The base protocol for optimistically sequenced merged mining is defined as follows:

1. **Initialization**. The auxiliary chain stores its current difficulty, target, and adjustment interval. The difficulty is set at the auxiliary chain's genesis and updated dynamically based on the submitted merged mined blocks.
2. **Auxiliary chain block production**. The Sequencer collects transactions and creates auxiliary chain blocks. If there are multiple Sequencers, they can employ a consensus protocol (e.g., round-robin, BFT protocols [8]) to coordinate block production.
3. **Prepare merged mining checkpoint**. Every N auxiliary chain blocks the Sequencer creates a checkpoint for Miners. This checkpoint references the N auxiliary chain blocks using a vector commitment scheme (e.g. Merkle tree containing the N auxiliary block hashes as leaves) or another efficient data structure.
4. **Merge mining.**
   (a) The Miner fetches the checkpoint from the Sequencer node(s) regularly, e.g., every 2 seconds.
   (b) The Miner verifies the signature of the signed blocks.

(c) The Miner includes the checkpoint into the coinbase transaction of the latest Bitcoin block mining template.
(d) The Miner performs PoW over this block template until a PoW solution is found that matches the required auxiliary chain difficulty
(e) The Miner sends the PoW solution (Bitcoin block) to the Sequencer.
(f) The Sequencer verifies that:
  – The PoW meets the required auxiliary chain difficulty.
  – The checkpoint is included in the coinbase transaction of the parent chain block.
  – The checkpoint includes only valid signed block hashes from the last mined block.

5. **Auxiliary chain finalizes N signed blocks**. The Sequencer signs and sends the Bitcoin block to other full nodes. Full nodes verify the PoW and the Sequencer signature.

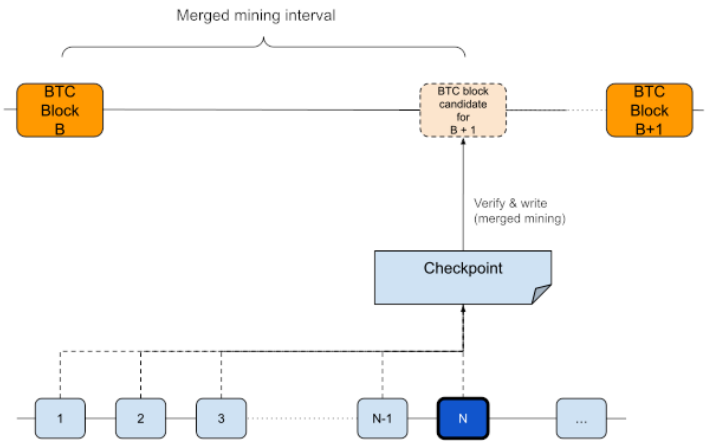

Fig. 1. Visualization of the OptiMine. The Auxiliary chain PoW difficulty targets a merge mined block to be found approximately every N auxiliary blocks. Merge mined blocks include the the checkpoint finalizing the set of auxiliary blocks signed by the Sequencer since the last merge mined block. Merge mined Bitcoin blocks can but do not have to become part of the Bitcoin main chain.

Auxiliary Chain Consensus

The auxiliary chain nodes will follow the heaviest chain (in terms of PoW) of finalized blocks. Thereby, the auxiliary chain deems the latest valid mined block as the canonical one. If no merged mined checkpoints are created for a predefined

period ("mining timeout"), e.g., a few hours, the Sequencer nodes will stop producing new signed blocks. This is considered a critical failure and will require mitigation (extensions discussed below).

Parent chain forks are ignored: the auxiliary chain only considers the PoW performed (implicitly) over the auxiliary block hashes.

4 Security

Sequencer Failures

We differentiate between Byzantine and Liveness failures of the Sequencer:

- **Byzantine failures**. Sequencers produce invalid blocks or equivocate by signing more than one block at the same block height. In this case, the Miners will reject the signed block(s) produced by the Sequencer. The auxiliary chain cannot achieve finality and should eventually stop producing blocks entirely.
- **Liveness failures** are caused by Sequencers failing to sign or broadcast blocks for a prolonged period of time. If the liveness failure extends beyond the mining timeout, finality cannot be reached.

In both cases, Miners will ensure that the auxiliary chain cannot be finalized. Hence, OptiMine protects against finalizing the auxiliary chain in case of Sequencer failures.

Miner Failures

Miners, in turn, can also cause Byzantine and Liveness failures:

- **Byzantine failures**. Miners can include or exclude signed blocks as part of the checkpoint as well as submitting mined blocks with insufficient difficulty. This would be rejected by the Sequencer when verifying and signing the submitted parent block.
- **Liveness failures**. If Miners fail to submit a valid parent block within the timeout period, the auxiliary chain will halt. This causes a liveness failure for the entire auxiliary chain.

Byzantine failures can be caught by honest Sequencers. However, Miner liveness failures will lead to the auxiliary chain stopping producing blocks and require out-of-band mitigation. It is worth mentioning that as long as a single miner is honest and online the chain will continue producing blocks, although edge cases may require additional handling (e.g. PoW difficulty adjustment intervals).

5 Applications

OptiMine can be used to add Bitcoin security to both standalone sidechains and rollups deployed on other L1 networks.

Sidechains

A standalone sidechain exhibits an independent block production mechanism. While the exact implementation depends on each sidechain's block production rules, handling of disputes between the sidechain nodes (Sequencers) and merged miners applies to all designs alike.

If for some reason, e.g., due to a consensus conflict between miners and sidechain nodes, no PoW solutions are submitted withing the timeout period, sidechain Sequencers must implement a conflict resolution protocol.

- **Wait**. Sequencers may wait until a valid parent block is submitted to continue block production.
- **Fork**. If no valid parent block has been submitted after a prolonged time, Sequencers may choose to fork the chain to continue without the merged mining, falling back to the vanilla sidechain consensus protocol.

Overall, the merged mining technique for sidechains requires opt-in and sufficient incentives for Miners to continue to operate the merged mining process. Equally, Miners can introduce liveness failures by stopping merged mining, which will require out-of-band resolution.

5.1 Rollups

In the case of rollups, settlement happens on a chain (e.g. Ethereum) other than the parent chain (e.g. Bitcoin). For example, by introducing optimistically sequenced merged mining, an Ethereum rollup can inherit Bitcoin PoW security.

We differentiate between "soft" and "hard" commitment deployments of OptiMine on rollups:

- **"Soft" commitment (App-level)**. Smart contracts deployed on the rollup can verify the presence of the last mined block and decide how to react to a safety or liveness failure. For example, apps could halt their operation if no PoW has been submitted for a pre-defined period, implementing a custom, extra security layer on top of the rollup.
- **"Hard" commitment (Rollup-level)**. The valid submission of the PoW, signature, and vector commitment becomes part of the L1 verification through either fraud or validity proofs. Specifically for optimistic rollups, the rollups state reverts if no valid PoW has been submitted for a prolonged time (as defined on the L1). For zk-rollups, the rollup state cannot be settled on the deploying L1 without a valid PoW. For example, if a zk-rollup settles every 6 hours on Ethereum, the zk-rollup needs to include a recent, valid PoW for the validity proof to be considered valid.

6 Extensions

Miner Validation and Emergency Fallback to Vanilla Merged Mining

This extension attempts to mitigate potential failures of the Sequencer. Instead of halting, the auxiliary chain switches into a recovery mode where Miners take

over block production, slowing down block times from, e.g., 2 seconds to 30 seconds. We differentiate between two types of failures:

- **Byzantine Failure**. Sequencer misbehaved by signing an invalid block or double-signing two blocks for the same height. In this case, the Miners will reject the sidechain block(s) produced by the Sequencer, and the network falls back to vanilla merged mining, i.e., Miners start self-producing blocks, and the consensus rules fall back to vanilla "longest chain".
- **Liveness Failure**. The Sequencer did not sign and broadcast a block template for a prolonged period. This would require all full nodes to regularly query the Sequencer for block templates (just like Miners would fetch the sidechain block templates for merged mining). Liveness failures are difficult to determine, and there may be edge cases. Miners would need to achieve consensus on whether the liveness failure is permanent before returning to vanilla merged mining. This is best achieved by long time-outs in practice (e.g., if standard block times are 30 seconds, the maximum timeout could be 15-30 minutes). If the Sequencer returns online, the system can recover and switch back to the sequenced merged mining model - e.g., when the network produces a merged mined block signed by the Sequencer.

We identify the following requirements to ensure correct operation of the emergency fallback mechanism:

- Miners must run full nodes on the auxiliary chain in parallel to the Sequencer.
- Auxiliary chain full nodes must implement the fallback mechanism as a custom consensus rule.

Securing Multiple Auxiliary Chains

Collected transaction fees across auxiliary chains vary. Thus, fee variance might lead Miners to ignore auxiliary chains with low incomes. To offset the fee variance of single auxiliary chains, checkpoints can be created across multiple chains.

To reduce overhead for Miners to run full nodes of each auxiliary chain, a new party can be introduced that we will term the aggregator. In the trusted setting, the aggregator would collect checkpoints from each auxiliary chain at regular intervals and provide the checkpoint to Miners. Sequencers of the auxiliary chains can query the aggregator to verify that their signed blocks are included in the checkpoint.

Trust in the aggregator can be removed by providing a ZK proof alongside the checkpoint that proves to both the Miners and the Sequencers that the checkpoint was created correctly.

Decentralizing Rollup Sequencers

When deployed on top of rollups, OptiMine can be used to decentralize the set of Sequencers by sampling candidates from Bitcoin miners. This can be achieved

by observing a sliding window (e.g. 2 weeks) of merge mined blocks, ranking miners by the number of mined blocks they generated, and electing the top N miners into the active Sequencer set for the next epoch (of e.g. 2 weeks). This approach requires miners to include a signature of their associated account on the rollup into the merge mined Bitcoin blocks to ensure authenticity of the sampled PoW solutions (i.e., to prevent miner A claiming to have mined blocks originally submitted by miner B). This model can be implemented in addition to a set of whitelisted Sequencers or combined with other Sequencer sampling protocols.

7 Conclusion

We introduced OptiMine, a new hybrid consensus technique that enables any sidechain or rollup connected to an L1 chain to receive Proof-of-Work security from Bitcoin Miners. Rollups and sidechains retain fast block production and reduce the trust in centralized Sequencers and block-producing nodes. OptiMine does not require any changes to Bitcoin and can be implemented on top of any sidechain, standalone chain or rollup.

References

1. Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J., Wuille, P.: Enabling blockchain innovations with pegged sidechains (2014), http://newspaper23.com/ripped/2014/11/http-_____-___-_www___-blockstream___-com__-_sidechains.pdf, accessed: 2016-07-05
2. Buterin, V.: A rollup-centric ethereum roadmap (2020), https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698
3. Collective, O.: Optimism docs (2024), https://docs.optimism.io/
4. Domo: brc-20 (2023), https://domo-2.gitbook.io/brc-20-experiment/
5. Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., Gervais, A.: Sok: Layer-two blockchain protocols. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security. pp. 201–226. Springer International Publishing, Cham (2020)
6. Jacobs, F.: Providing better confidentiality and authentication on the internet using namecoin and minimalt (apr 2014), https://raw.githubusercontent.com/FredericJacobs/safeweb/master/paper.pdf, accessed: 2014-04-02
7. Judmayer, A., Zamyatin, A., Stifter, N., Voyiatzis, A.G., Weippl, E.: Merged mining: Curse or cure? In: CBT'17: Proceedings of the International Workshop on Cryptocurrencies and Blockchain Technology (Sep 2017), https://eprint.iacr.org/2017/791.pdf
8. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. vol. 4, pp. 382–401. ACM (1982), http://people.cs.uchicago.edu/~shanlu/teaching/33100_wi15/papers/byz.pdf
9. Lerner, S.D.: Rootstock whitepaper (2019), https://rootstock.io/static/a79b27d4889409602174df4710102056/RS-whitepaper.pdf
10. Linus, R.: Bitvm: Compute anything on bitcoin (2023), https://bitvm.org/bitvm.pdf

11. O'Connor, R., Piekarska, M.: Enhancing bitcoin transactions with covenants. In: Financial Cryptography and Data Security (Apr 2017), https://fc17.ifca.ai/bitcoin/papers/bitcoin17-final28.pdf
12. Pass, R., Shi, E.: Hybrid consensus: Scalable permissionless consensus (Sep 2016), https://eprint.iacr.org/2016/917.pdf, accessed: 2016-10-17
13. Rodarmor, C.: Ordinal theory handbook (2024), https://docs.ordinals.com/