TL;DR: this note covers Taiko proving designs for theGrímsvötn testnet (Alpha-3) and Eldfell testnet (Alpha-4), briefly discussing its pros, cons, and open questions. We invite Taiko current provers and anyone potentially interested in participating as a prover to share feedback for current proving design and suggest any relevant improvements and alternatives. We'd also like to highlight that there is a grant category for alternative Proposer-Prover Tokenomics and Proof Markets, check it out if you might be interested in working on it.

EIP-1559 style proving design (Grímsvötn L2, Alpha-3 testnet)

TLDR: the "most efficient" prover wins. Below we explain what exactly "the most efficient" means and what the game rules are.

How it works:

- Anyone can join as a prover and leave at any time;
- One proof should be confirmed for one "window." A "window" is a period of time in which multiple blocks are proposed. That is, multiple blocks are recursively proven together to lower the (i) per block on-chain tx confirmation cost and (ii) per block verification cost;

[

ProofWindow

878×457 10.8 KB

](https://global.discourse-cdn.com/standard17/uploads/taiko/original/1X/80ca18be7950cfe66a4dfc2010f1cfae7e06e8d7.png)

- Any prover can submit a proof for any amount of blocks at any time. However, there is a target window, n, that is considered to be the most efficient in terms of the trade-off between low enough costs for users and fast enough withdrawals;
- There is a target reward, x, that is paid to the prover if they confirm the proof exactly at the target window, t = n. If the proof is confirmed before

the target window, the paid reward will be lower

than the target reward. Conversely, if the proof is confirmed after

the target window, the paid reward will be higher

than the target reward;

[

Target reward

855×466 7.37 KB

](https://global.discourse-cdn.com/standard17/uploads/taiko/original/1X/d2530deadce8aac3c975780f9d4c51b6f4134a54.png)

- A target reward is defined based on the historical reward values and is adjusted after each window depending on the proof confirmation time (that is where the EIP-1559 mechanism is applied). That is, if in window i the proof was confirmed before the target window n, the target reward for window i + 1 will be lower than the target reward for window i. On the opposite, if in window i the proof was confirmed after the target window n, the target reward for window i + 1 will be higher than the target reward for window i.

[

Target reward

826×174 6.53 KB

](https://global.discourse-cdn.com/standard17/uploads/taiko/original/1X/b91e155a0307029b4779b72a0adacfba8c845ab2.png)

Conclusion

With the EIP-1559 approach to the prover market design, the most efficient point is reached. That is, the proof cost for users

(implicitly being a share of tx cost) is at the lowest possible level, while prover rewards are still lucrative enough for provers to participate.

To be efficient within this design, a prover should be able to find an optimal trade-off point between (i) confirming the proof as late as possible (to get the higher reward) and (ii) confirming the proof earlier than all other provers. Otherwise, even if the prover wins one block, at the same, time it might lose several blocks. So, his resource spending for these several blocks will be larger than the rewards earned for one proven block.

Disclaimer:

it is worth noting that to confirm the proof as early as possible is NOT

an optimal strategy for the prover. And even though de-facto the fastest proof is accepted, confirming all proofs as fast as possible decreases the rewards making it unreasonable for provers (but beneficial for users).

With the current design, it's probable that (i) there will be one prover who'll generate all proofs at the lowest price or (ii) provers will collaborate off-chain (similar to mining or staking pools) and the "prover union" will generate all proofs at the lowest price. In both cases, a new prover can jump in anytime and start generating the proofs at even lower price. That will provide the lowest proof cost possible for users and prevent the chain from artificial prover rewards "pumping."

Staking based proving design (Eldfell L3, Alpha-4 testnet)

TLDR: one prover is pseudo-randomly chosen for each block. The probability of being chosen depends on its stake and expected reward.

- Anyone can join as a prover and leave at any time. In the exit case, the prover is immediately no longer assigned as a prover. The stake can be withdrawn in a week.
- For each block, the prover is chosen randomly from the 32 top provers with the largest weight;
- Prover weight W is calculated based on the stake A and expected reward per gas R:

[

Rewards Formula

1842×162 8.97 KB

](https://global.discourse-cdn.com/standard17/uploads/taiko/original/1X/eac92bc8d3d410309d17f213e3d170cf61833ce1.jpeg)

- The prover weight reflects its probability to be chosen;
- The current fee per gas F is calculated based on historical values and is supplied by the core protocol. It is modified every time a block is verified by its assigned prover within the proving window;
- Three other parameters are individual for each prover and are claimed while joining the proving pool:
- Number of Taiko Tokens to stake A;
- The expected reward per gas, R, is limited to (75% – 125%) * F range. If the R claimed by the prover is below or above this range, R will be automatically fixed at 75% * F or 125% * F, respectively;
- Maximum capacity C reflects how many blocks the prover can prove in parallel. While the prover generates proof, his capacity C is decreased by one (to reflect his free capacity at the current moment).

When the assigned prover submits the proof OR if proofTimeWindow elapses and another prover proves the block, when that block gets verified, the capacity C of the initially assigned prover is increased by one;

- If the assigned prover fails to prove the block within a specific time window, it is slashed;
- If the prover failed to prove the block or there is no available prover at the moment to be assigned, any prover can jump in and prove the block. Such a block is considered an "open block";
- If the block was proven, the prover reward is R * gasUsed.

Conclusion

With staking-based proving design, the problem when several provers generate proofs for the same block in parallel but only one proof is accepted (the fastest one) is solved.

Under this design, the provers with the highest stakes and the lowest expected rewards have higher chances of being chosen. It encourages multiple provers to stick around so that they could prove each others blocks when necessary.

However, as there are only 32 provers in the prover set for one slot, one can't say that this design is perfectly inclusive, especially when it comes to individual and solo provers. This issue can be mitigated through the secondary proof market that will lower the entry barrier for individual and solo participants (e.g., bringing together individuals who only have proving hardware (no tokens) and those who only have tokens (no hardware).

Contribution call

- We encourage anyone to submit issues to improve the existing staking-based tokenomics or share any feedback and ideas at this forum;
- We are also looking forward to new innovative designs and developments for a proof market infrastructure that will empower small and solo provers. Check the Taiko Grant Program, sections "Proof Markets" and "Alternative Proposer-Prover Tokenomics";
- We plan to hold an open research discussion about current proving design and opportunities for further improvements at the 11th of August. Reach out to lisa@taiko.xyz if you'd love to take part in.