Review of the ProgPoW Stakeholder Response

The Ethereum Cat Herders as part of gauging community response to ProgPoW engaged with several key community stakeholders and asked them three specific questions regarding ProgPoW. Below is the collective, generic response from these stakeholders.

Question 1: What questions would you like to see be answered as part of the technical audit of ProgPoW being conducted by the Ethereum Cat Herders?

1. The expected effects of ProgPoW on the security of Ethereum vis-a-vis:
    a. Security of the algorithm
        i. Does ProgPoW enforce the same (or better) cryptographic strength already experimented with ethash?
        ii. Has the ProgPoW code been tested/studied enough to confirm it does not hide exploitable flaws?
    b. Attack surface
    c. Cost of 51% attack
    d. Other security risks that may result from a change from Ethash to ProgPoW
        i. Ethash uses kecak-f1600 with padding and ProgPoW uses keccak-f800 without padding. How does this affect the security of the algorithm?
        ii. The FNV hashing algorithm used for DAG ordering in ProgPoW is non-cryptographic could this be a potential attack surface?

2. ProgPoW meeting the claimed goal of ASIC resistance

    Which version of ProgPoW is going to be audited?
    Does ProgPoW serve its stated purpose?
    Does the specification match the reference implementation?

    a. Known methods to speed up the calculation of the hash function
        i. Is the achievable speedup from a dedicated hardware implementation of ProgPoW small?
        ii. Are there ways to cheat the randomness part with an FPGA?
        iii. What is the expected performance difference between GPUs and ASICs?
        iv. What are the recommended ProgPoW parameters to make difference between GPUs and ASICs minimal?
        v. Can ASICs implement the "naive" ProgPoW implementation (the same that CPUs are implementing and in which there is no program precompiled but the spec is "interpreted")?

b. Length of time it would it take to create a ProgPoW ASIC (if R&D begins immediately)

    i. How plausible is to create specialised hardware for ProgPOW within a few months, which would outperform GPU by 50% in terms of hashes per watt, adjusted by the ratio in the cost of production to the cost of purchasing a GPU?

    ii. If you were a hardware designer, how would you go about designing an ASIC for ProgPOW, what would be the first experiments to run?

    iii. Is it feasible for an ASIC manufacturer to produce an ASIC for ProgPoW? Meaning, what would be the estimated ROI, and how realistic would it be to compete with Intel and AMD on their 'home turf'?

    iv. Assuming it might be not technically impossible to build an ASIC for ProgPoW are there any authoritative opinions which can help settle how hard could it be to build one? (this involves the evaluation of the costs and timings of the whole industrialization process for such a specialized device)

    v. Tunables embedded in ProgPoW can be considered enough to slow down any incoming ASIC wave during the (unfortunately) slow transition to full PoS?

c. Expected efficiency gains from the first generation of said ASICs

3. Identify any potential advantages or disadvantages that ProgPoW would present in comparison to Ethash in terms of changes to the network, “fair mining” and evaluate any potential uneven distribution

    a. GPU Advantage: Resulting in manufacturer centralization due to hashrate advantage

        i. Are there any pieces of evidence ProgPoW code has been deliberately written to favor Nvidia vs AMD ? When comparing different vendors is the retail price criteria taken as a base parameter?

        ii. Do we have any benchmark study which appeases the rumors about unfair advantages reachable by one chip vendor or another?

        iii. What is the expected performance difference between AMD and Nvidia cards?

        iv. What are the recommended ProgPoW parameters to make a difference between AMD and Nvidia cards minimal?

        v. How much time will be needed to carry out an official benchmarking test on behalf of tunable settings proposal depicted here https://medium.com/@ifdefelse/progpow-progress-da5bb31a651b?

            In (partial) reply to the question, from a volunteer tester: https://medium.com/@infantry1337/the-miners-benchmark-progpow-e79cab6eabc3

b. Light Clients:
   i. What will be the impact of longer validation times on light clients?
   ii. Have any developer teams for light clients already reported this or been questioned on this?

c. Block Time Validation
   i. What would the impact be of reduced hashrate be on block validation times?
   ii. How much time will it take for network difficulty to adjust?
   iii. Would it be necessary to adjust the diff multiplier at the block number decided for ProgPoW?

4. Impact on the Clients and Core Devs
   a. Considering that at this very moment only two major clients support progpow (geth and parity by the means of still unmerged PRs) what is the timeframe for all currently active clients to get compliant?
   b. Is it possible to implement ProgPoW on mainnet with an earlier fork other than Istanbul?

Question 2: What, in your opinion, are the major Pros and Cons of the ProgPoW proposal?

Pros

1. General Lower-Barrier of Entry into Mining Ethereum:
   a. Keeping GPU mining industry in business for a bit longer, if this is the goal.
   b. If it functions as designed, it offers a way to continue to favor mining Ethereum with general-purpose hardware and resists centralization due to effective monopolies by dedicated hardware manufacturers.
   c. ProgPoW has been studied (and implemented) to expressly slow-down the wave of specialized mining hardware. While it may sound appealing the idea economies of scale to help reduce overall energy consumption, those, on the other hand, have the purpose to concentrate in few powerful hands the mining service thus progressively locking the development of software to the needs of hardware. We've seen this before with hash wars on Bitcoin chain and its splits supported by major ASICs manufacturers/vendors.
   d. The general public around the world can obtain GPU cards in large quantities;
      i. without import restrictions,
      ii. without special deals with the manufacturer,
      iii. without having to preorder and wait for future batches to be produced
2. The belief that no ProgPoW ASIC would be produced. It took asic-manufacturers years to start producing ethash, with moderate success. The thought is that gains are even smaller on ProgPoW, and thus not even worth doing.
3. Keeping the original GPU miners is good for Ethereum

   a. Keep existing decentralized user base of primarily GPU miners, who are supportive of ethereum and the goal to eventually transition to PoS.
   b. There is a "social" impact: the first adopters in crypto are PoW miners. From enthusiasts/hobbyists to small/mid-sized mining facilities, individuals, more than enterprises, have contributed more to the success, so far, of Ethereum.

4. ProgPoW adoption is a signal of openness to all players in the mining industry and levels the field.
5. ProgPoW will recall back a consistent part of the "dormant" hashpower many GPU miners have.
6. ProgPoW will keep the network distribution safe.
7. The idea of changing part of the algorithm every 50 blocks.

Cons

1. Share validation time. Ethash share validation time is a flaw of Ethash compared to other modern PoW algorithms like Equihash or Cuckoo cycle. ProgPoW is a step back here with validation time even more than Ethash has.
2. If Linzhi 1400 MH/s for 100 W ASIC is not fake, they probably found some way to avoid DAG generation like mentioned here: https://bitcointalk.org/index.php?topic=1716584.msg46521718#msg46521718 The problem is ProgPoW has exactly same DAG.
3. ProgPoW’s own analysis shows 1.2 ASIC performance gain over GPU. Meanwhile, GPUs are used mainly for graphics calculations and have ALUs tuned for floating point performance. ProgPoW does not use floating point calculations at all so ASIC gain coefficient can be bigger. 1.2 efficiency gain looks unproven.
4. There is no paper (for example, in arxiv.org or iacr.org) describing the algorithm with formal proofs. Like MTP, Cuckoo cycle or Equihash papers.
5. It is harder to verify (2x), it makes mining more costly (more energy spent per GPU), which makes GPU mining less competitive (it increased the advantage of people who can buy electricity for cheaper), and potentially more centralized.
6. It introduced appetite for perpetual tweaks in Proof Of Work, which creates a lot of distraction for Ethereum 1x project.
7. The largest risk is the risk of fragmentation inherent in a hard fork that implements changes that aren't universally as an uncontroversial technical upgrade.
8. Adds tech debt, all clients will need to have both Hashimoto and ProgPoW verification implemented in order to sync (even just header-sync).
9. Adds a little bit of verification delay.
10. Makes block verification on mobile even heavier.
11. Verification is slowed, asymmetry between proof and verification is a key value of PoW.
12. Changing the PoW algorithm creates winners and losers.
13. Bricking 100k Bitmain E3 owners is a bad thing for Ethereum and could lead to a backlash.
14. GPU is fork technology and inherently less secure than ASICs.

15. Large GPU miners with special deals with NVIDIA/AMD gain a cost advantage that leads to centralization.
16. ProgPoW is a brainchild of a known team who have been involved in questionable businesses in the past. There is a fear that this team may benefit by using their knowledge of ProgPoW to assist a specific GPU company with tuning their GPUs to mine more efficiently than the other.
17. Going ahead with ProgPoW will divide the community (it's already happening).
18. Allowing ProgPoW to occur will be similar to the Monero ASIC resistance hard forking experience and this needs to be studied so as not to repeat the same experience.
19. ASIC resistance is an ongoing battle. ProgPoW is not meant to solve the problem permanently: its primary purpose is to delay the ASIC wave through the time needed to transition to PoS.
20. ProgPoW is not perfect and ongoing bugs will lead to a development cost and might require some future changes.
21. Hardware development is meant to catch-up as long as manufacturers can foresee enough profit from huge investments.
22. The “hashrate for hire” problem poses a greater risk of a 51% attack if the hashrate is reduced.

Question 3: In your opinion, should ProgPoW be implemented into Ethereum?

Yes: 43%

Barring any untoward discoveries, or clear signs that it would lead to a contentious split, yes.

No: 29%

It should either be improved or Ethereum should adopt another approach like EthashV2 (https://eips.ethereum.org/EIPS/eip-969) changing the algorithm slightly every four months to maintain ASIC resistance until PoS.

Yes & No: 14%

ProgPOW can be implemented, but it should not be activated. If it is, have an option in the clients to oppose it and provide replay protection in case there is a chain split.

Indifferent: 14%