TLDR

: We present a minimal randomness beacon using a Verifiable Delay Function (VDF). We argue for the safety and liveness of the RANDAO + VDF scheme and conclude with some discussion points.

Thanks to @prateek and @mihailobjelic for feedback.

Construction

Assume a global clock and split time into contiguous 8-second blocks and 128-slot epochs. Each epoch i

produces 32 bytes of (biasable) entropy e_i

to which correspond 32 bytes of (unbiasable) randomness r_i

. In a recursive fashion, the beacon chain proposers of epoch i

(one per slot) are sampled using past randomness r_j

(i.e. j = i - N

for some suitable constant N

).

Biasable entropy (RANDAO)

Every beacon chain proposer is committed to 32 bytes of local entropy. (In practice a chain of commit-reveals is setup with a hash onion for validator registration.) Beacon chain proposers may reveal their local entropy by extending the canonical beacon chain with a block. Honest proposers are expected to keep their local entropy private until their assigned slot.

The beacon chain maintains 32 bytes of onchain entropy by XORing the local entropy revealed at every block. Denote by e_i

the onchain entropy at the last slot of epoch i

.

Unbiasable randomness (VDF)

Let D

be a (verifiable) delay function which takes 32-byte inputs x

and integer time parameters T

, and which returns 32-byte outputs y = D(x, T)

. We assume commodity VDF hardware can evaluate outputs no slower than A_{max}

times what an attacker can do with proprietary hardware. That is, A_{max}

is the maximum attacker advantage.

We fix the time parameter T

so that the commodity hardware takes A_{max}

epochs (i.e. 8\times 128\times A_{max}

seconds) to evaluate outputs and we define r_i = D(e_i, T)

. Anyone can run the commodity VDF hardware to compute and broadcast the r_i

(with the evaluation proofs allowing fast verification) to become a so-called evaluator. Honest beacon chain proposers are expected to include the r_i

(and corresponding evaluation proofs) onchain.

Safety and liveness arguments

We make three further assumptions:

- Majority validator honesty

: We assume 2/3 of the validators are honest. (This assumption is also made for sharding.)

- Minority validator liveness

: We assume up to 3/4 of the honest validators are offline, possibly permanently. (This assumption also captures validator censorship, e.g. networking DoS.)

- Altruistic evaluator

: We assume at least one altruistic evaluator runs the commodity VDF hardware and broadcasts $r_i = D(e_i, T)$ (with an evaluation proof) at the end of epoch $i + A_{max}$.

Safety argument

We argue that the randomness r_i is unbiased. Using a recursion argument, let’s assume that r_j (where $j = i - N$) is itself unbiased. (The first few values of r_i can be set using a PoW blockhash.) Then using the validator honesty and liveness assumptions the probability that epoch i has no honest proposer is bounded above by $\big(1-\frac{2}{3}\times\frac{1}{4}\big)^{128} < 2^{-33}$, i.e. is negligible.

Now assume an attacker runs proprietary hardware that is A_{max} times faster than the commodity hardware. Such an attacker can compute $D(e_i, T)$ for varying e_i in exactly one epoch. Because the honest proposer in epoch i reveals his local entropy no earlier than epoch i, and the attacker grinding opportunity on e_i is limited to epoch i, grinding is impossible and r_i is unbiased.

Liveness argument

The altruistic evaluator assumption guarantees that the randomness r_i (with an evaluation proof) corresponding to epoch i is available offchain by the end of epoch $i + A_{max}$. As argued above, each epoch has at least one honest proposer who can guarantee inclusion of r_i by the end of epoch $i + A_{max} + 1$.

Justifying the assumptions

Safety argument

The key assumption for the safety argument is the existence of commodity VDF hardware that is no slower than A_{max}

times what proprietary hardware can achieve. The current plan is for the Ethereum Foundation and Filecoin (and possibly others like Chia and Solana) to jointly develop a VDF ASIC to get a reasonable A_{max}

.

Rough estimates suggest that a budget of $20m-$30m is sufficient for the commodity ASIC to support A_{max} = 10

for 5 years. This ASIC would be replaced within 5 years as part of an upgrade to a quantum-secure VDF. (Similar to BLS signature aggregation, the initial VDF scheme is not quantum-secure.)

Liveness argument

The key assumption for liveness is the existence of an altruistic evaluator. Assuming the commodity hardware is widely distributed for free across the Ethereum community, it is reasonable to assume at least one altruistic evaluator despite the lack of in-protocol rewards.

Indeed, similar to how 14,000+ nodes are operated without in-protocol rewards, we can expect the community (enthusiasts, developers, investors, dApp operators, the Ethereum Foundation, etc.) to altruistically run VDF evaluators around the world. Similar to enthusiasts overclocking CPUs and GPUs, we can expect some VDF ASICs to get overclocked.

Discussion

Other randomness constructions

- Biasable randomness

: Most known randomness beacons are biasable via last revealer(s) attacks. This includes PoW (Bitcoin, Ethereum 1.0), RANDAO + VRF (Algorand, Cardano), RANDAO + PVSS (Tezos), RANDAO + low-influence functions (Polkadot).

- Dfinity’s randomness

: The threshold relay scheme stands out as not being biasable. Unfortunately, the beacon can stall if even a minority (e.g. 15%) of honest players go offline. A design goal for Ethereum 2.0 is to survive WW3 making strong liveness non-negotiable.

- VDF-based randomness

: The only known randomness beacons that have both strong safety and strong liveness use a VDF.

Practical considerations

- Baseline security

: If the VDF breaks down completely (e.g. a quantum computer makes outputs computable with no delay) the randomness beacon falls back to RANDAO security.

- Validator/evaluator decoupling

: Validators only need to verify VDF outputs using evaluation proofs, hence do not need to be evaluators. Evaluators do not need to be registered or collateralised, hence do not need to be validators.

- Ethereum 2.0 roadmap

: The Ethereum 2.0 roadmap is independent of the VDF ASIC. A VDF upgrade strengthening RANDAO can come with phase 1 (sharding data layer), phase 2 (sharding state layer), or later. The development of commodity hardware will likely take at least 18 months.

- ASIC power usage

: It is estimated that each evaluation core in the VDF ASIC would consume less than 10 Watts even under extreme voltage and temperature. Assuming 10,000 active VDF cores running at 10 Watts, that is ~23,000 times less power consumption than today’s PoW mining in Ethereum 1.0.

- Bandwidth overhead

: The beacon chain must include one VDF output and evaluation proof per epoch. With the Wesolowski VDF (using a 2048-bit RSA modulus, or a 2048-bit class group discriminant) that corresponds to 512 bytes of overhead per epoch, i.e. 0.5 bytes per second.

- Computation overhead

: Verifying a VDF output against an evaluation proof takes ~1 ms of CPU time for the suggested VDF (see above). That corresponds to ~30 seconds of CPU time overhead per year. No significant network-level DoS vector is introduced thanks to the low verification overhead per VDF output.

Optional infrastructure

- Inclusion rewards

: It is particularly easy to give an inclusion reward (e.g. 0.1 ETH) to the first beacon chain proposer that includes randomness r_i

. This reward would help prevent rational proposers slipping into laziness. It may also incentivise sophisticated validators to overclock the commodity VDF ASIC as an indirect evaluation reward (see below).

- Evaluation rewards

: At the cost of added protocol complexity, VDF evaluators can receive direct in-protocol rewards. Watermarking the evaluation proofs to evaluator identities would be required (and is easily done). Note that if evaluation rewards are non-existent or small (e.g. $3K per day, ~$1m per year) the development of a proprietary VDF ASIC seems unlikely.

- Difficulty mechanism

: At the cost of added protocol complexity, a difficulty scheme (e.g.see discussion here) can be added to dynamically adjust the time parameter T

over long timescales (e.g. decades) without making use of hard forks every few years. Until a quantum-secure VDF is implemented a difficulty adjustment mechanism is likely unnecessary.

Benefits of unbiasability

- Consensus parameters

: With unbiasable randomness the relevant consensus parameters (e.g. honesty and liveness assumptions, committee sizes and thresholds) do not have to include safety margins to mitigate bias. Alternatively, the safety margins can stay, making the design stronger.

- Formal analysis

: With unbiasable randomness the analysis of the consensus layer is greatly simplified allowing for rigorous safety proofs and formal verification. When validators can bias the randomness complicated games arise and it is unclear whether such games are fully understood.

- Randomness opcode

: dApps (e.g. lotteries) get programmatic access to unbiasable randomness via an opcode. Replicating an equivalent layer 2 randomness beacon without protocol-level support is hard.

- Competitiveness

: Unbiasable randomness at the protocol-level keeps Ethereum 2.0 competitive with other third-gen blockchains (e.g. Dfinity and Filecoin).