TLDR

: We suggest a random beacon scheme where committees of size n

generate random numbers if k

participants participate correctly. It is in a similar vein to Dfinity's random beacon (without use of BLS) and has the same message complexity of k

messages per beacon output.

Part 1: A single random output

For clarity we first show how k

-of-n

participants generate a single output in three phases. In part 2 we combine the phases for messaging efficiency.

- Phase 1—ephemeral identities

: Every participant is committed to an ephemeral secret key sk_i

and shares the corresponding ephemeral public key pk_i

.

- Phase 2—encrypted shares

: Every participant uses his secret key sk_i

to deterministically build a degree-(k-1)

polynomial P_i

by interpolating the points (0, sk_i), (1, H(sk_i || 1)), ..., (k-1, H(sk_i || k - 1))

. The polynomial P_i

is then used to create n

shares P_i(1), ..., P_i(n)

. Finally, the shares P_i(j)

are encrypted with respect to the public keys pk_j

, and the encrypted shares are committed to and shared publicly. We say that a participant "commits correctly" if both the public key pk_i

and the encrypted shares are well-formed relative to the secret key sk_i

.

- Phase 3—reveal

: All n

participants are invited to reveal their ephemeral secret key sk_i

. When a participant reveals his secret key sk_i

anyone can check if that participant committed correctly. (Participants that do not commit correctly can be slashed.) Assuming k

participants reveal their secret key and committed correctly, if it possible to determine which of the participants who did not reveal their secret key committed properly, and extract the secret key for those who did commit properly. Indeed, k

of the n

encrypted shares of any participant can be decrypted with the revealed secret keys, which is enough to recover a candidate polynomial P_i

and a corresponding secret key sk_i

, and check that the participant committed correctly. We now define the random output to be the sum of the secret keys sk_i

for which the corresponding participants committed correctly.

Part 2: The random beacon

Similar to RANDAO, every participant is committed to a hash onion and we use the preimages as ephemeral secret keys. In order to combine the above three phases into one message, the reveal phase is used to “refill” ephemeral public keys and encrypted shares for future beacon rounds.

For maximum messaging efficiency and to cater for non-participation in some of the rounds we have a buffer of publicly shared future commitments (of size, say, 10) which is refilled appropriately in the reveal phases.

Discussion

One benefit of the leaderless approach is that there is no leader who has monopoly knowledge over the next beacon output and can singlehandedly abort (a weakness of the RANDAO plus a k

-of-n

committee approach).

By using a quantum-secure encryption scheme (e.g. supersingular elliptic curve Diffie-Hellman?) we can mimic Dfinity’s random beacon with the same optimal messaging complexity of k

messages per round.