Tail-Hammer: Optimized Statistics for Anonymous Committees and Applications

Bernardo David[1], Lucia Lavagnino[2], Elena Pagnin[2], and Paul Stankovski Wagner[3]

[1] IT University of Copenhagen, bernardo@bmdavid.com
[2] Chalmers University of Technology and University of Gothenburg, {luciala, elenap}@chalmers.se
[3] Lund University, paul.stankovski_wagner@eit.lth.se

Abstract. Techniques to randomly select sets of anonymous parties are ubiquitous in efficient and adaptively secure consensus protocols, as well as in Multi-Party Computation in the YOSO model, where each round is executed by a different random anonymous committee. Anonymous committee selection aims at randomly selecting a set of n parties (the committee), where at most t parties are corrupted (except with negligible probability), drawing from a population of $N \gg n$ parties with at most T corrupted parties. Additionally, each party knows (and can prove) if they belong to the committee, but ignores other members' identities.

A very common and efficient instantiation of anonymous committee selection is to select parties according to a VRF output, this however, leads to committees of probabilistic size (n behaves as a Binomial random variable). Despite wide adoption, only [7] (Blum et al., CCS23) provides an analysis of VRF-based probabilistic anonymous committee selection that estimates the size of committees. This analysis relies on lose bounds (Chernoff) and approximations (Poisson).

In this work, we revisit Blum et al.'s estimates and derive accurate closed-form formulas (based on a tight Binomial approximation), as well as an efficient high-precision library called Tail-Hammer for computing exact parameters. Notably, Tail-Hammer identifies smaller committee sizes (approximately -25% on average) than [7] for the same security level, leading to improved efficiency in protocols relying on random committee selection, also when anonymity is not needed. Our analysis applies to committee selection techniques that employ unbiased (uniformly random), or bounded-bias randomness, to both synchronous and asynchronous communication settings, and it can account for inactive parties.

As a new application, we present a verifiable consistent broadcast protocol that leverages quorums in anonymous committees to achieve efficiency without requiring threshold signatures.

1 Introduction

A popular paradigm to improve scalability in consensus is selecting *small random committees* to execute different parts of a protocol [23,31,34], rather than requiring all parties to execute the entire protocol. Works extending this approach to selecting *anonymous* committees have achieved not only scalable but also *adaptively secure* consensus protocols [7,16,18,22]. In a nutshell anonymous committee selection is like a random committee selection that does not publicly disclose the identity of committee members. The adversary therefore does not know which parties to corrupt until they act, a feature needed in adaptive security. More recently, the same approach has yielded similar improvements in Multi-Party Computation (MPC). Protocols in the Fluid MPC model [6,17,20,24,33] employ small committees for improved scalability, while protocols in the YOSO model [5,10,12,13,26] achieve both scalability and adaptive security by executing each round with a new small anonymous random committee.

When instantiating protocols that leverage randomly selected committees, one of the main scalability challenges is identifying *the smallest possible* number of parties to execute each step of the protocol. When selecting a committee of a known fixed size n, this is achieved by determining the smallest n for which it is guaranteed that at most t parties are corrupted, except with negligible probability, given that committees are randomly selected from a population of N parties out of which at most T are corrupted. This is the case when committees are publicly selected (without anonymity) with the help of a randomness beacon, as in [23,31,34].

Selecting committees of optimal (*i.e.*, smallest) sizes becomes considerably harder in settings where the committee size itself is probabilistic. This is usually the case for protocols that require anonymous committees, e.g., Algorand [7,16], Dfinity [30] Ouroboros [31] and follow-ups [3,22], Snow White [18], Can a Public Blockchain Keep a Secret? [5], YOSO [26] and follow-ups [10,12,13]. While techniques for randomly selecting anonymous committees of a prescribed size are known, they suffer from prohibitive communication [2,13,27] and computational [8,9,14,15,19] overheads. Hence, such protocols are usually instantiated using anonymous committee selection schemes based on Verifiable Random Functions (VRF), as proposed in [5,7,16,18,22].

In the aforementioned VRF-based committee selection schemes, each of the N parties is selected as a committee member with a certain probability p, so that the expected value for the committee size n is pN, and the expected value for the committee corruption ratio $\frac{t}{n}$ is $\frac{T}{N}$. Previous approaches [5,16,18,22] derive the committee size n by upper-bounding the number of corrupted committee members t assuming that $n \in [pN - \Delta, pN + \Delta]$ parties are selected at random, and then determining p so that n stays within this range. Blum et al.'s analysis of Algorand [7] is the only work that provides concrete parameter estimates for the various committees and quorums instantiated in the protocol. The *quorum* is a value Q such that receiving messages from at least Q committee members guarantees that a sufficient number of honest parties have acted. While committees with more than Q members are still selected, protocols leveraging these committees can safely ignore (and discard) additional messages received from the remaining committee members after the first Q messages are received. Consequently, all efficiency and security estimates can be expressed in terms of the concrete and fix quorum value Q, rather than the variable expected size of the committee. Blum et al.'s analysis relies on Chernoff bounds and Poisson approximation, which the authors themselves acknowledge to be loose in the regime of cryptographic negligibility, thereby raising concerns regarding their qualitative significance. In particular, [7] concludes with the open problem of deriving tighter security bounds without modifying the expected committee sizes.

We address this challenge via a new analytical technique that yields strictly tighter bounds than those derived in [7] and enables efficient, exact parameter computation through our *Tail-Hammer* library. Table 1 compares Blum et al.'s estimates (cf. Table 1 in [7]) with the values produced by Tail-Hammer at the same target bit-security level. From Table 1 it is evident that the looseness of the Chernoff bounds and Poisson approximations leads to unnecessarily large parameters. Our techniques show how committee and quorum sizes can be reduced to approximately 75% on average, maintaining the same security level. We hope these improvements will motivate future work to abandon the use of Chernoff bounds and instead to adopt Tail-Hammer for exact estimation of committee sizes in probabilistic anonymous committee selection.

Committee Type	C (expected committee size)			Q (quorum size)		
	Algorand [7]	Tail-Hammer	$\frac{\text{Tail-Hammer}}{\text{Algorand}}$	Algorand [7]	Tail-Hammer	$\frac{\text{Tail-Hammer}}{\text{Algorand}}$
soft-vote*	n/a	**4136**	n/a	n/a	**3161**	n/a
soft-vote	2990	**2372**	79.3%	2267	**1786**	78.8%
cert-vote	1500	**1110**	74%	1112	**812**	73%
next-vote	5000	**3632**	72.6%	3838	**2767**	72.1%
late-vote	500	**398**	79.6%	320	**247**	77.2%
redo-vote	2400	**1786**	74.4%	1768	**1298**	73.4%
down-vote	6000	**4388**	73.1%	4560	**3305**	72.5%

Table 1: Comparison of expected committee sizes (C) and estimated quorums (Q) between Algorand [7, Table 1] and Tail-Hammer. Under Algorand bit-level security requirements, Tail-Hammer reduces both C and Q of each committee type. For instance, for next-vote, the committee size can be reduced from 5000 to 3632, maintaining the same targeted bit-level security. soft-vote* revises soft-vote as explained in Section 4.2.

1.1 Contributions

Our main contribution is a novel security analysis that applies to random, and to anonymous committees selection techniques returning committees of probabilistic size. Our analysis justifies working with a relatively small subset of committee members, a quorum, while achieving the same cryptographic security level as with the full committee. In detail, we make the following contributions.

We introduce the *"split-bell" approach* (Section 3), which treats malicious, honest, and inactive parties separately. From this approach, we can statistically identify a *quorum* setting the threshold for the minimal number of committee members required to act in order to guarantee a desired fraction of honest participants. Our approach is both intuitive and powerful, as it directly connects the probability of bad events to the tail areas of binomial distributions. We analyze quorums for subcommittees that contain at least one honest party or an honest majority in the presence of ideal reliable broadcast resources (Section 3.2) and in the absence thereof (Section 3.3).

We introduce *tail measures* (Section 4.1), which serve both to accurately estimate bad events in different settings and to provide a precise and unified security metric.

We provide *closed-form formulas* (Section 4.3) to estimate the quorum and the selection probability, given any desired bit-level security and the population composition (fractions of actively malicious and inactive parties in the system).

We provide an open source high-precision Python library, *Tail-Hammer* [1], to compute exact tail areas of binomial distributions, which are essential for identifying optimal quorum values. We present indicative figures obtained using Tail-Hammer and compare these against state-of-the art estimates from [7] (Table 1) as well as our tight theoretical approximation formulas (Table 4).

Finally, we further demonstrate the usefulness of quorums in random anonymous committees by constructing a simple *authenticated verifiable consistent broadcast protocol* (Section 5). Our protocol is both adaptively secure and efficient without requiring complex setup assumptions, such as the distributed key generation needed for threshold encryption or signatures.

Limitations. Our analysis considers only probabilistic committee selections for random and for anonymous committees in which each party has *the same probability* p of being selected and that yield committees of probabilistic size. Generalizing this to different probabilities (as in the proof-of-stake setting) is not straightforward. However, our approach can be applied in scenarios where parties are naturally chosen with equal probability (*e.g.* MPC) or in scenarios where it is possible to "emulate" stake via virtual parties, *i.e.* having each party control a number of virtual parties proportional to their stake. For an improved analysis of committee size in stake-based selection (without quorums), see [25].

Extensions. Although we present our results in the context of anonymous committees, the analysis also applies to *non-anonymous* settings. Notice that using a simple and cheap committee selection scheme based on VRFs is also advantageous in applications where the committee's identity is publicly known, only requiring an extra step for each selected committee member to make themselves known. For example, applying our results to Gearbox [23] renders the protocol robust against randomness bias and inactive parties, thus obtaining smaller quorums and eliminating the requirement for an unbiasable randomness beacon.

1.2 Our Techniques

Our starting point are probabilistic committee selection techniques where the expected committee size is distributed as a binomial random variable with parameters N (the total number of parties) and p (the probability of each party being selected). Our objective is to identify a fixed minimum number Q (the *quorum*) of committee members that guarantees a desired honesty threshold, with suitably high probability. The fixed quorum can conceptually replace the probabilistic committee size, simplifying protocol analysis and implementation.

Let $0 \leq f_M < 1/2$ denote the fraction of malicious parties running anonymous committee selection. Parties can be (conceptually and statistically) split into two categories; $T = f_M N$ malicious and $H =$

$(1 - f_M)N$ honest ones. Analyzing the selection process of honest and malicious parties separately leads to two *independent* random variables with distributions $\mathcal{B}_T \sim B(T, p)$ and $\mathcal{B}_H \sim B(H, p)$. Figure 1 exemplifies this *split-bell approach* to understanding the composition of anonymous committees drawn from a two-class population.

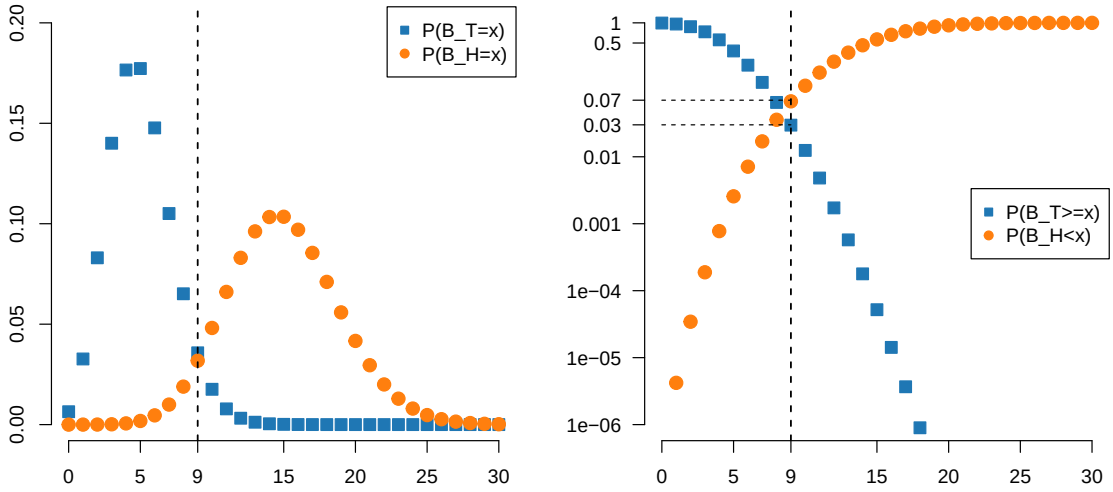

Fig. 1: Illustration of the "split-bell" approach using two binomial distributions, with toy parameters chosen for illustrative purposes; malicious parties $\mathcal{B}_{250}$ (blue squares) and honest parties $\mathcal{B}_{750}$ (orange dots). $N = 1000$ users, $H = 750$, $T = 250$, corruption fraction $f_M = 0.25$ and selection probability $p = 0.02$. Plots are cropped at $x = 30$. The dashed lines at $x = 9$ mark the quorum Q. The left figure displays the probability mass function of the two distributions. The right figure plots the cumulative distribution function (cdf) of $\mathcal{B}_H$ in orange dots, and the complement of the cumulative distribution function (1-cdf) of $\mathcal{B}_T$ in blue squares. In this example, $Q = 9$ gives $\Pr(\mathcal{B}_T \geq Q) < \Pr(\mathcal{B}_H < Q) \leq 0.07 \sim 2^{-3.8}$.

Our search for exact quorums stems from observing that as long as $T \neq H$, $\mathcal{B}_H$ and $\mathcal{B}_T$ have different means and variances. In particular, they are drawn from two distinct "bell-shaped" distributions with non-colliding peaks which offer a range in which to look for a value Q for which both the left-most bell's right-tail, and the right-most bell's left tail are suitably "small". Such a quorum Q simultaneously bounds the risk of too many corrupted committee members and too few honest ones (see Figure 1 for a graphical representation). Whether the bounds are concretely useful (in terms of bit security level) depends on the shapes of the bells (i.e., the system parameters H, T and p).

More generally, our results are derived from one overarching methodology, which we refer to as the split-bell approach. Let $X_1, X_2 \in \mathbb{N}$ with $1 < X_1 < X_2$. Consider two functions $\mathsf{f}_1, \mathsf{f}_2 : \mathbb{N} \setminus \{0\} \to \mathbb{N}$. The role of these two functions is to formalize, in mathematical terms, the quorum conditions that one aims to achieve. For example, in the case of at least one honest party, both f_1 and f_2 are the identity function, whereas in the honest-majority case, f_1 corresponds to the function $\lfloor \frac{\cdot}{2} \rfloor$. To build intuition, set $X_1 = T$, $X_2 = N$, and $\mathsf{f}_1(x) = \mathsf{f}_2(x) = x$. Next, consider the following probabilities identified by the areas under the tail of the binomial distributions (for $Q \in \mathbb{N}$):

$$\Pr(\mathcal{B}_{X_1} < \mathsf{f}_1(Q)) \qquad (1a) \qquad\qquad \Pr(\mathcal{B}_{X_2} \geq \mathsf{f}_2(Q)) \qquad (1b)$$

We can interpret (1a) as the probability of there being "enough honest parties" in the committee, hence *safety*; and (1b) as the probability of having enough committee members selected, hence *liveness*. With this

in mind we look for a suitable number Q for which both probabilities are high, leading to the fact that, with high probability, a randomly-chosen subset of Q-many committee members has "enough honest parties", e.g., honest majority. Hence this approach allows for intuitively balancing security and efficiency as well as for efficient quantitative evaluation through our Tail-Hammer code [1]. The split-bell approach we just described appears to be novel in the literature on anonymous committee selection.

Finally, we note that when $X_2 = N$ and $\mathsf{f}_2(x)$ is the identity function, (1b) implies the quorum be smaller than the actual committee size. This yields two key advantages. First, it reduces the concrete communication complexity from proportional to pN (which is the expected committee size) to just $Q < pN$. For example, for committees that require at least one honest party, $Q = pN\sqrt{f_M}$ (see the bounds in Section 4.3 and concrete optimal values in Table 4). Second, it gives a minimum number of outputs needed to advance to the next protocol phase, which is particularly valuable in asynchronous settings.

2 Preliminaries

Building Blocks. We use standard EUF-CMA digital signatures [29], Key Evolving Signatures (KES) [4] and Verifiable Random Functions (VRF) [32] with unpredictability under malicious key generation [22,28].

Consistent Broadcast. Consistent Broadcast [11] allows for a potentially corrupted sender to broadcast a message such that not all honest parties are guaranteed to output but all honest parties that do output are guaranteed to output the same message. When the sender is honest, all honest parties are guaranteed to output the sender's message. Such a scheme is considered verifiable if it is possible to produce a message that causes any honest party to output the same message as the other honest parties. We will construct a simple adaptively secure authenticated verifiable consistent broadcast scheme based on random anonymous committees. In Appendix A, we recall the definition of consistent broadcast with such properties.

Anonymous Committee Selection. An anonymous committee can be selected from a set of parties using a VRF [16,18,22], among other techniques. In VRF-based committee selection, each party determines if they are part of a committee by evaluating the VRF on a nonce r from a random beacon using their secret key sk_i, $\mathsf{VRF.Eval}(\mathsf{sk}_i, \mathsf{r}) \to (y_i, \pi_{y_i})$ and checking that $y_i < \theta_i$, where θ_i is a threshold that determines the probability of selecting the party for the committee. Later on, selected parties can prove they are committee members by revealing (y_i, π_i), which other parties can check to satisfy $y_i < \theta_i$ and $1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_i, \mathsf{r}, y_i, \pi_{y_i})$. Given that VRF keys are fixed before r is known and their output has at most bounded bias, this procedure yields a committee of certain expected size and number of honest parties that cannot be manipulated by the adversary, except with negligible probability.

In this work, we consider that all parties are elected with equal probability, *i.e.*, θ_i is the same for all parties. Since VRF outputs are independent across parties (each party has its own key pair) and uniformly pseudorandom, all selection outcomes are identically distributed as independent Bernoulli random variables. Summing up the trials over all N parties yields a Binomial random variable $B(N, p)$ with $p = \theta/2^\ell$ where 2^ℓ is the range of the VRF outputs. Hence we can reason about the distribution of committee sizes using binomial distributions.

3 Quorums in Probabilistic-size Anonymous Committees

Given an anonymous committee selected at random (*e.g.* via the VRF-based schemes of [16,18,22]), we here explain how to identify a *quorum* Q, *i.e.*, a threshold such that when Q members are sampled from the anonymous committee, the probability that this subset contains the required number of honest parties is sufficiently high. Specifically, we consider quorums for at least one honest party, and for honest majority.

3.1 Analysis Setup

The committee selection process is modeled as a random variable that appoints each party as a committee member with probability p. This intuitively emulates a local coin flip, where the coin lands on "heads" with probability p. In particular, each party's selection outcome is an independent and identically distributed Bernoulli random variable with a given success probability p. Consequently, the total number of parties selected into the committee follows a Binomial distribution $B(N, p)$, where N is the total number of eligible parties.

Our analysis utilizes a standard result in probability theory:

> the sum of independent Binomial random variables with the same success probability p is itself a Binomial random variable.

Therefore, by (conceptually and statistically) splitting the population running the selection into two disjoint sets of cardinalities $T = f_M N$ and $H = (1 - f_M)N$, (where T denotes the maximum number of parties assumed to be malicious, $H = N - T$ denotes the honest parties, and $0 \leq f_M < 1/2$ is the assumed fraction of maliciously corrupted users) we obtain

$$B(N,p) = B(H+T,p) \stackrel{d}{=} B(H,p) + B(T,p) \tag{2}$$

where $\stackrel{d}{=}$ means to have the same distribution function. The latter equation provides a means to analyze malicious and honest parties' selection independently. In particular, $B(T, p)$ identifies a well-defined discrete probability distribution that is different, in shape, from $B(H, p)$ for $T \neq H$. In short, our analysis is based on this shape difference.

In what follows, we consider any probabilistic committee selection that chooses parties independently with a fixed probability p, and such that the size of the committee is variable (e.g., VRF-based cryptographic sortition). We let $\mathcal{B}_T \sim B(T, p)$, $\mathcal{B}_H \sim B(H, p)$, $\mathcal{B}_N \sim B(N, p)$ denote the random variables describing the number of malicious, honest, and total committee members selected in the selection process. In this setting, the exact realizations of the random variables $t \leftarrow \mathcal{B}_T$, $h \leftarrow \mathcal{B}_H$ and $n = t + h \leftarrow \mathcal{B}_N$ are not immediately observable by any party. However, it is expected that $t \sim pT$ and $h \sim pH$ (and that the committee has size $n \sim pN = p(T + H)$). Even without knowing the exact values of n and t, we can proceed to identify a fixed value $Q < pN$ for which both the probability of there being at least Q parties in the committee is large (say, $\geq 1 - 2^{-b}$) and the probability of there being *too many* malicious parties among any Q selected members is small (say, $< 2^{-b}$). For efficiency, we are interested in the smallest Q that satisfies these constraints. For security, we are interested in values of b in the range of cryptographic security parameters.

3.2 Quorum Analysis

Quorums for "at least one honest party". We work with the split-bell approach (see (1a) and (1b)), where we set $X_1 = f_M \cdot N = T$ for the left bell (which describes the probability distribution of selecting malicious parties); and $X_2 = N$ for the right bell (which describes the probability distribution of the anonymous committee selection process overall), and $\mathsf{f}_1(x) = \mathsf{f}_2(x) = x$ (identity function). We then ask that:

$$\text{both } \Pr\big(\mathcal{B}_N < Q\big) \text{ and } \Pr\big(\mathcal{B}_T \geq Q\big) \text{ be small enough}$$

to simultaneously satisfy liveness (the probability of selecting a large enough committee) and safety (the probability of selecting enough honest parties). The first condition states that the probability that the total number of selected parties (the size of the committee) is smaller than the quorum Q is small. The second condition states that the probability that the number of selected malicious parties exceeds the quorum Q is also small. Together, these two conditions ensure that a subset of Q random committee members must contain at least one honest party with high probability, and that the quorum lies in the interval $pT < Q < pN$.

Quorums for "honest-majority". We again use the split-bell approach (see (1a) and (1b)) with $X_1 = f_M \cdot N = T$, $X_2 = N$, $\mathsf{f}_1 = \lfloor x/2 \rfloor$, $\mathsf{f}_2(x) = x$, but ask

$$\text{both } \Pr\big(\mathcal{B}_N < Q\big) \text{ and } \Pr\big(\mathcal{B}_T \geq \left\lfloor \tfrac{Q}{2} \right\rfloor\big) \text{ be small enough.}$$

The first condition enforces the quorum be smaller than the committee size, so there will always be Q-many committee members (liveness) except for a negligible probability. The second condition ensures Q be large enough, so that the probability of choosing a subset of Q-many committee members containing more than $Q/2$ malicious parties is small enough (safety). Together the constraints imply honest majority in any subset of Q members of the committee.[4]

3.3 Quorum Analysis Over Asynchronous Networks

Since Q is strictly smaller than the expected committee size, there exist multiple subsets of size Q within the committee. We refer to these as *shadow subcommittees*. Avoiding shadow subcommittees with different views is easier assuming reliable broadcast: all parties broadcast and, after all broadcast instances terminate, define the quorum as the first Q committee members whose broadcast terminated successfully in lexicographical order on the VRF output. Asynchronous networks, on the other hand, do not assume any setup, and consider an adversary who may arbitrarily delay, or drop, messages exchanged by honest parties. These conditions may prevent honest parties from identifying the same subset of parties forming the quorum.

To avoid shadow subcommittees over asynchronous networks, Q cannot be chosen too small – it must be at least half of the expected committee size; otherwise, the adversary could temporarily sustain the coexistence of two or more subcommittees of size Q. Even if Q is large enough to prevent the existence of two *non-overlapping* shadow subcommittees, malicious parties may "double dip" and spawn two independent quorums of parties as shown in Figure 2.

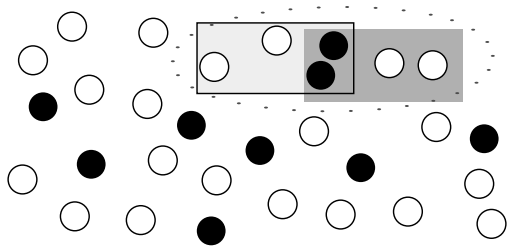

Fig. 2: An illustrative example of shadow subcommittees for $N = 30$, $T = 9$ ($f_M = 0.3$), $Q = 4$, and $p = 0.2$. White (resp. Black) circles depict honest (resp. corrupted) parties. The gray dotted line confines the randomly selected anonymous committee of size $n = 6 = pN$. Within this committee, two shadow subcommittees of size $Q = 4$ are represented by gray shaded rectangles.

Quorums for "at least one honest party". Over asynchronous networks, to prevent inconsistent views from different shadow subcommittees, Q needs to be larger than the expected number of malicious committee members, and also of half of the expected number of honest members in addition to it. Therefore we set $X_1 = \frac{(1+f_M)}{2}N$, and keep $X_2 = N$, $\mathsf{f}_1(x) = \mathsf{f}_2(x) = x$:

$$\text{both } \Pr\big(\mathcal{B}_N < Q\big) \text{ and } \Pr\big(\mathcal{B}_{T+\frac{H}{2}} \geq Q\big) \text{ be small enough.}$$

The first condition sets an upper bound on Q, and guarantees that it is always possible to find Q-many committee members. The second states that the malicious committee members should not be able to spawn two coexisting shadow subcommittees of size Q by exploiting the existence of "too many" honest members (~~and the byz~~antine behavior of malicious ones).

[4] To see this consider $\Pr\left(\mathcal{B}_N \geq Q\right) = 1 - \epsilon_1$ and $\Pr(\mathcal{B}_T < \lfloor Q/2 \rfloor) = 1 - \epsilon_2$. Since $\mathcal{B}_N = \mathcal{B}_T + \mathcal{B}_H$ the first inequality can be written as $\mathcal{B}_T + \mathcal{B}_H \geq Q$ which is equivalent to $\mathcal{B}_H \geq Q - \mathcal{B}_T$. Using the second inequality ($\mathcal{B}_T < \lfloor Q/2 \rfloor$) we obtain $\mathcal{B}_H > Q - \lfloor Q/2 \rfloor$, i.e., $\mathcal{B}_H \geq \lceil Q/2 \rceil$ where the inequalities hold with probability at least $1 - \frac{\epsilon_1}{1-\epsilon_2}$.

Quorums for "honest majority". Over asynchronous networks, to prevent honest parties from receiving inconsistent views from different shadow subcommittees, Q must be chosen sufficiently close to the actual committee size. In this way, even though corrupted parties may participate in multiple subcommittees due to their Byzantine behavior, there are not enough honest parties available to populate two distinct quorums. This can be enforced precisely as in the case of "at least one honest party" and imposing the additional "honest majority" constraint (a third bell in the picture):

$$\Pr(\mathcal{B}_N < Q), \quad \Pr(\mathcal{B}_{T+H/2} \geq Q), \quad \text{and} \Pr(\mathcal{B}_T \geq \left\lfloor \frac{Q}{2} \right\rfloor) \quad \text{be small}$$

The first condition guarantees liveness. The second rules out shadow subcommittees. Even if all malicious members collude and attempt to split the honest parties into two groups, neither subcommittee can independently reach Q-many members. The third condition prevents any randomly chosen subcommittee of size Q from being dominated by a dishonest majority and thus guarantees safety.

3.4 Handling Inactive Parties

Although in practice one may not be able to a priori label parties as actively malicious, actively honest, or inactive, the committee selection procedure can be statistically modeled as drawing from the different classes independently.

The split-bell approach can be naturally extended to account for inactive parties. Suppose that at most f_I of the total population is expected to be inactive (but not malicious) during committee selection. By replacing N with the actual number of active parties $(1-f_I)N$ in our analysis, we can incorporate the impact of inactivity while preserving the same reasoning about quorum thresholds and tail probabilities (which will be affected by a smaller active population). Accounting for inactivity through an explicit assumption allows us to model modern global systems more accurately, since parties may be inactive for various reasons ranging from time-zone differences to local electricity shortages. Moreover, this framework leaves open the choice of treating inactive parties either as "inactive honest" or as "corrupted but silent". In the latter case, our analysis shows that inactive parties cause less harm to security than actively corrupted ones and can be used to quantify the different levels of damage depending on whether malicious parties are active or not.

4 Estimating and Computing Exact Quorums

4.1 Tail Measures for Weighting Bad Events

Section 3 is intentionally vague on how to measure security (which encompasses safety and liveness). All that is asked for is that the areas of certain binomial tails "be small". Even at an intuitive level, it is clear that this property is necessary for security: the tails are defined in such a way that if they had "large" probability, then some bad event is likely to happen, e.g., randomly selecting a subset of committee members of size Q without the desired honesty threshold. Since every case handles more than one tail, and tails may differ in size, estimating the bit-security level directly from the split-bell approach cannot be done in a unique and straightforward way.

The aim of this section is to distill meaningful *tail measures*, that concretely define the probability of *failing to meet* the target property promised by the quorum, i.e. any random subset of Q-many committee members has sufficiently-many honest parties, except with a small probability.

A *tail* is a function $\tau : \mathbb{N} \setminus \{0\} \to [0, 1]$ that maps positive integers (the candidate quorum values q) to probabilities. The tail function $\tau^X_{\mathsf{dir}}(q) = \Pr(\mathcal{B}_X \quad \mathsf{ineq} \quad q)$ is defined by the following parameters:

- The probability $p \in]0, 1]$ of any party being selected as committee member, which we keep implicit since it is fixed in our analysis;
- The population size X;
- The tail direction $\mathsf{dir} \in \{\mathsf{left}, \mathsf{right}\}$;

Name of measure	Quorum definition	Bit level
Max of tails	$Q = \min\Big\{ q \in \mathbb{N} \;\Big|\; \tau^{X_1}_{\mathsf{right}}(q) < 2^{-b} \;\wedge\; \tau^{X_2}_{\mathsf{left}}(q) < 2^{-b} \Big\}$	b
Sum of tails	$Q = \min\Big\{ q \in \mathbb{N} \;\Big|\; \tau^{X_1}_{\mathsf{right}}(q) + \tau^{X_2}_{\mathsf{left}}(q) < 2^{-b} \Big\}$	b
Safety-over-liveness	$Q = \min\Big\{ q \in \mathbb{N} \;\Big|\; \tau^{X_1}_{\mathsf{right}}(q) < \tau^{X_2}_{\mathsf{left}}(q) \;\wedge\; \tau^{X_2}_{\mathsf{left}}(q) < 2^{-b} \Big\}$	b
Independent tails	$Q = \min\Big\{ q \in \mathbb{N} \;\Big|\; \tau^{X_1}_{\mathsf{right}}(q) < 2^{-b_1} \;\wedge\; \tau^{X_2}_{\mathsf{left}}(q) < 2^{-b_2} \Big\}$	$b = \min\{b_1, b_2\}$

Table 2: Some tail measures with corresponding quorum definition and bit level. Here $X_1, X_2 \in \mathbb{N}$, and $b_1, b_2, b \in \mathbb{R}$.

– An inequality symbol $\mathsf{ineq} \in \{<, \leq, \geq, >\}$, where ineq is $<$ or $\leq$ for $\mathsf{dir} = \mathsf{left}$; and $>$ or $\geq$ for $\mathsf{dir} = \mathsf{right}$.

Note that in our setting, tail values correspond to probabilities of *bad events* for security. With this in mind, we introduce *tail measures*, i.e., a handful of different ways to weight the probability of bad events defined by tails. The purpose of the tail measures is to capture the different security requirements imposed by the protocol. Table 2 displays the *quorum for each tail measure*, as the minimum value Q such that the tail measure is satisfied. For simplicity we use only tails with strict inequalities, and weight up to two bad events. Extension to more than two tails, or more bad events, can be done easily, whenever meaningful.

It is easy to see that both safety-over-liveness and sum of tails are more conservative measures than max of tails. Following a conservative approach, our quorum approximation formulas in Section 4.3 use the safety-over-liveness tail measures, which guarantees security (both liveness and safety) with overwhelming probability; but liveness is allowed to have more failures than safety.

4.2 Comparison with previous works

We compare the security levels and expected committee sizes obtained with our technique to those available for Algorand in [7]. This comparison demonstrates that:

1. Chernoff bounds on the binomial distribution are not sufficiently tight for cryptographically interesting values (see Table 3);
2. Recomputing Algorand committee parameters using our Tail-Hammer library leads to smaller quorum and expected committee sizes (see Table 1).

Algorand [16] achieves consensus through a Byzantine agreement protocol which, for scalability, uses seven distinct types of committees: propose, soft-vote, cert-vote, next-vote, late-vote, redo-vote, and down-vote (see Table 1 in [7]). Each committee has a specific role, expected size, bit security level, and tail measure. Our analysis excludes the propose committee, as it does not require a quorum.

In the notation of the *split-bell approach*, [7] determines the quorum Q by applying two distinct tail measures. Given two binomial distributions $\mathcal{B}_{X_1}$ and $\mathcal{B}_{X_2}$, $X_1 < X_2$, the quorum Q is the value such that the right tail of $\mathcal{B}_{X_1}$ is smaller than a threshold determined by the desired security level with high probability (safety: $\tau^{X_1}_{\mathsf{right}}(\mathsf{f}_1(Q)) \leq 2^{-b_1}$); and that the left tail of $\mathcal{B}_{X_2}$ is smaller than a second threshold determined by a corresponding security level only with small probability (liveness: $\tau^{X_2}_{\mathsf{left}}(\mathsf{f}_2(Q)) \leq 2^{-b_2}$). Their statistical analysis can be described using the *independent tails measure* with the quorum Q defined as

$$Q = \min \Big\{ \; q \in \mathbb{N} \;\Big|\; \tau^{X_1}_{\mathsf{left}}(\mathsf{f}_1(q)) \leq 2^{-b_1} \quad \wedge \quad \tau^{X_2}_{\mathsf{right}}(\mathsf{f}_2(q)) \leq 2^{-b_2} \Big\}$$

where b_1 and b_2 differ by orders of magnitude (see e.g., [7]). Table 3 summarizes the various requirements of b_1 and b_2 for each committee, assuming a population of $N = 10^{12}$ participants, where an adversary can adaptevely corrupt at most $f_M = 0.2$ fraction of parties running the anonymous committees. In general, b_{X_1} is a cryptographic security level, while $0 < b_{X_2} \ll b_{X_1}$ is acceptable for liveness.

In Blum et al.'s analysis of Algorand [7] the condition used to prevent the formation of *shadow committees* in soft-vote is $\tau_{\mathsf{right}}^{H+2T}(2Q) \leq 2^{-b_1}$, i.e., $X_1 = H + 2T$ and $\mathsf{f}_1(x) = 2x$. As explained in Section 3.3, however, to prevent shadow subcommittees with conflicting views, the quorum size Q must exceed the number of malicious committee members T plus half of the honest ones, hence $\tau_{\mathsf{right}}^{\frac{H}{2}+T}(Q) \leq 2^{-b_1}$. Notably $\tau_{\mathsf{right}}^{H+2T}(2Q) \leq 2^{-b_1}$ is not equivalent to $\tau_{\mathsf{right}}^{\frac{H}{2}+T}(Q) \leq 2^{-b_1}$. The issue is that $2\mathcal{B}_X \stackrel{d}{=} \mathcal{B}_{2X}$ does **not** hold (what holds is $\mathcal{B}_{2X} \stackrel{d}{=} \mathcal{B}'_X + \mathcal{B}_X$ assuming two independent identically distributed random variables, see Equation (2)). In the present case, however, the same binomial random variable is summed twice, which implies that the terms are necessarily dependent.

Tables 1 and 3 contain figures for soft-vote*, which denotes the committee parameters under the correct condition $\tau_{\mathsf{right}}^{\frac{H}{2}+T}(Q) \leq 2^{-b_1}$; and soft-vote which uses configuration of [7] satisfying $\tau_{\mathsf{right}}^{H+2T}(2Q) \leq 2^{-b_1}$.

In Algorand statistical analysis [7], the authors estimate values for the binomial distribution using Chernoff bounds for the tail, and approximate the Binomial with a Poisson distribution. Table 3 shows that this approximation is not tight. In Table 3 we use the same parameters for probability and committee size as in [7], and compute exact bit levels using Tail-Hammer. Every committee yields higher bit levels for safery than what estimated via bounds and approximation; except for soft-vote*, where the revised analysis shows that the values in [7] only reach 88 bits of safety. Table 3 shows that the bit levels for liveness remain unchanged due to their low magnitude; at this scale, the effect of the approximation introduces less variance than the direct application of the binomial function.

Committee Name	C	Q	b_{X_1} (safety)		b_{X_2} (liveness)	
			Algorand	Tail-Hammer	Algorand	Tail-Hammer
soft-vote*	2990	2267	n/a	**88**	n/a	**7.7**
soft-vote	2990	2267	128.2	**171**	7.7	**7.7**
cert-vote	1500	1112	673	**936**	7.7	**7.7**
next-vote	5000	3838	2401	**3360**	7.7	**7.7**
late-vote	500	320	166	**225**	16.0	**16.0**
redo-vote	2400	1768	1064	**1473**	12.2	**12.2**
down-vote	6000	4560	2827	**3942**	12.1	**12.1**

Table 3: Comparison between safety and liveness bit-levels for different committees as estimated in [7] for Algorand, and as computed with Tail-Hammer. For instance, for committee cert-vote, [7] estimates $\Pr(\mathcal{B}_{X_1} \geq Q) \leq 2^{-673}$ for the small bell, while the actual value attained is $\Pr(\mathcal{B}_{X_1} \geq 2Q) \leq 2^{-936}$.

4.3 Quorum Approximation Formulas

We present a practical method to compute p and Q given the total number of parties running the anonymous committee scheme N, the fraction of malicious parties f_M and of inactive ones f_I, and the desired bit security level b.

For simplicity, we present the methodology only for the max of tails measure, as the other cases can be derived from this blueprint. Our formulas are derived by approximating the binomial distribution with a Gaussian (Normal distribution $\mathcal{N} = \mathcal{N}(0,1)$),[5] and using *quantiles* ϕ_b of the normal distribution (i.e. $\Pr(\mathcal{N} \leq \phi_b) = 2^{-b}$). The values obtained through the formulas require adjustments to account for the approximation error introduced by fitting a normal distribution to the binomial model. How these approximations affect accuracy is discussed in detail in Appendix D. The formulaic result generally deviates by fewer than 100

[5] These approximations correspond to $\Pr(\mathcal{B}_X \leq x) \sim \Pr\left(\mathcal{N} \leq \frac{x-pX}{\sqrt{Xp(1-p)}}\right)$ using the fact that $\mathcal{N}(0,1) = (\mathcal{N}(\mu,\sigma^2) - \mu)/\sigma$.

units from the actual quorum. Table 4 displays values computed with the formulas that we explain in the remainder of this section.

For a two-bell system of the form $\begin{cases}\Pr(\mathcal{B}_{X_1} \geq \mathsf{f}_1(Q)) \\ \Pr(\mathcal{B}_{X_2} < \mathsf{f}_2(Q))\end{cases}$ we set:

$$\Pr\left(\mathcal{N} \geq \frac{\mathsf{f}_1(Q) - pX_1}{\sqrt{X_1 p(1-p)}}\right) = \Pr\left(\mathcal{N} < \frac{\mathsf{f}_2(Q) - pX_2}{\sqrt{X_2 p(1-p)}}\right) = 2^{-b}.$$

For the three-bell system $\begin{cases}\Pr(\mathcal{B}_{X_1} \geq \mathsf{f}_1(Q)) \\ \Pr(\mathcal{B}_{X_2} < \mathsf{f}_2(Q)) \\ \Pr(\mathcal{B}_{X_3} \geq \mathsf{f}_3(Q))\end{cases}$ we set:

$$\Pr\left(\mathcal{N} \geq \frac{\mathsf{f}_3(Q) - pX_3}{\sqrt{X_3 p(1-p)}}\right) = \Pr\left(\mathcal{N} < \frac{\mathsf{f}_2(Q) - pX_2}{\sqrt{X_2 p(1-p)}}\right) \leq \Pr\left(\mathcal{N} \geq \frac{\mathsf{f}_1(Q) - pX_1}{\sqrt{X_1 p(1-p)}}\right) = 2^{-b}.$$

We can impose equalities due to approximating discrete binomials with continuous normal distributions.

Formulas for p and Q in the "at least one honest party" case. From Section 3.2 we know that $\mathsf{f}_1(x) = \mathsf{f}_2(x) = x$ in this case. By the properties of the normal distribution, the area of the left tail, i.e., $\Pr\left(\mathcal{N} < \frac{Q - pX_2}{\sqrt{X_2 p(1-p)}}\right)$, is equal to the area of the right tail when the points are symmetric with respect to the y-axis. This provides a way to relate the quorum Q to the probability p of being selected as committee member. Specifically, we get these formulas (see Appendix C, for the details): $p = \frac{\phi_b^2}{(\sqrt{X_2} - \sqrt{X_1})^2 + \phi_b^2}$, $Q = p\sqrt{X_1 X_2}$.

Combining these results with the ones from Section 3.2, we can substitute X_2 with N; and X_1 with either $T = f_M \cdot N$ (reliable broadcast) or $T + H/2 = (1 + f_M) \cdot N/2$ (asynchronous networks). As a result, the formulas imply that $Q = pN\sqrt{f_M}$ in the reliable broadcast case, and $Q = pN\sqrt{(1 + f_M)/2}$ in asynchronous networks. These formulas show that while the quorum scales proportionally to the expected committee size (which is pN) it does so by a shrinking factor $\sqrt{f_M}$ (resp. $\sqrt{(1 + f_M)/2}$).

Formulas for p and Q in the "honest majority" case. In reliable broadcast networks, one can apply the same methodology as the previous case using $\mathsf{f}_1(x) = \frac{x}{2}$ and $\mathsf{f}_2(x) = x$. Simple calculations lead to these formulas:
$p = \frac{\phi_b^2}{\left(\frac{X_2 - 2X_1}{\sqrt{X_2} + 2\sqrt{X_1}}\right)^2 + \phi_b^2}$, $Q = 2p\sqrt{X_1 X_2}\frac{\sqrt{X_1} + \sqrt{X_2}}{2\sqrt{X_1} + \sqrt{X_2}}$. The ratio between the quorum size and the expected committee size is $\frac{Q}{p \cdot N} = 2\frac{\sqrt{X_1}}{\sqrt{X_2}}\frac{\sqrt{X_1} + \sqrt{X_2}}{2\sqrt{X_1} + \sqrt{X_2}}$. Substituting X_2 with N, and X_1 with $f_M \cdot N$ (from Section 3.2) the ratio can be expressed as $2\sqrt{f_M}(\sqrt{f_M} + 1)/(2\sqrt{f_M} + 1)$. The previous formula shows that this ratio depends solely on the malicious threshold The asynchronous networks case is more contrived as it involves a three-bell system with $\mathsf{f}_1(x) = \frac{x}{2}$, $\mathsf{f}_3(x) = \mathsf{f}_2(x) = x$. Simple calculations (see Appendix C , for the details) lead to: $p = \frac{\phi_b^2}{\left(\frac{\sqrt{X_2 X_3} - 2X_1}{\sqrt{4X_1}}\right)^2 + \phi_b^2}$, $Q = p\sqrt{X_2 X_3}$, where ϕ_b is the right quantile of the normal distribution (i.e. $\Pr(\mathcal{N} \geq \Phi_b) = 2^{-b}$) Substituting X_1 with $f_M \cdot N$, X_3 with $\frac{1 + f_M}{2} \cdot N$, and X_2 with N, we get $\frac{Q}{pN} = \frac{\sqrt{1 + f_M}}{\sqrt{2}}$.

4.4 Numerical Assessment

Table 4 collects quorum sizes and selection probabilities for the cases of at least one honest party and honest majority. Exact values are computed using our Tail-Hammer library [1] and closely match the theoretical estimates. The values display a nearly constant ratio between the quorum Q and the expected committee size pN, as predicted by our formulas. For example, assuming reliable broadcast and requiring quorums with at least one honest party (resp. honest majority) at 128-bit security for an active population of size $N = 10^9$

with $f_M = 0.3$ malicious parties, the quorum Q is just 481 (resp. 3,940) while the expected committee size $C = pN$ is roughly 828 (resp. 4,820). When comparing Table 1 with Table 4, the latter exhibits seemingly inferior results. However, this discrepancy arises from the differing tail measures employed, rendering the values across these tables incomparable.

At least one honest party case

		Reliable Broadcast			Asynchronous Network		
b	N	10^9	10^5	10^4	10^9	10^5	10^4
128	C	**828** (833)	**822** (826)	**763** (769)	**4,540** (4,540)	**4,342** (4,349)	**3,117** (3,122)
	Q	**481** (484)	**477** (480)	**441** (445)	**3,688** (3,688)	**3,526** (3,532)	**2,523** (2,528)
60	C	**376** (377)	**374** (376)	**363** (363)	**2,055** (2,050)	**2,017** (2,012)	**1,703** (1,704)
	Q	**218** (219)	**218** (218)	**210** (210)	**1,669** (1,665)	**1,638** (1,634)	**1,381** (1,382)

Honest majority case

		Reliable Broadcast			Asynchronous Network		
b	N	10^9	10^5	10^4	10^9	10^5	10^4
128	C	**4,819** (4,814)	**4,589** (4,586)	**3,214** (3,212)	**4,819** (4,810)	**4,589** (4,589)	**3,214** (3,248)
	Q	**3,940** (3,936)	**3,750** (3,748)	**2,614** (2,612)	**3,940** (3,932)	**3,750** (3,750)	**2,614** (2,634)
60	C	**2,180** (2,180)	**2,129** (2,130)	**1,778** (1,780)	**2,180** (2,174)	**2,129** (2,128)	**1,778** (1,784)
	Q	**1,782** (1,782)	**1,740** (1,741)	**1,450** (1,452)	**1,782** (1,776)	**1,740** (1,738)	**1,450** (1,454)

Table 4: Optimal quorums Q and expected committee sizes C under the safety-over-liveness measure. Values in **bold** are computed with **Tail-Hammer**, values in parentheses are estimated via the formulas in Section 4.3. Bit security: b; total active parties: N; corruption tolerance: $f_M = 0.3$; selection probability: $p = \frac{C}{N}$.

5 Adaptively Secure Consistent Broadcast

We construct a protocol for Authenticated Verifiable Consistent Broadcast [11] that achieves adaptive security against Byzantine adversaries over asynchronous networks and efficiency without the need for complex setup such as distributed key generation for [11], which is based on threshold encryption. Our protocol builds on the construction in [11] but leverages quorums of anonymous committees for efficiency. The main idea is to select a small set of parties with honest majority and have them sign the broadcast input message along with a proof they have been selected. Once the party who initiated the broadcast collects quorum-many valid signatures and proofs, it relays these to all of the other parties, who output the message.

The VCBC Protocol. Our VCBC protocol (Protocol 1) assumes a PKI registering all parties' public keys and a random beacon, which we later realize. The protocol is parameterized by a threshold value θ (for selecting committee members), and the quorum size Q, which are determined according to the fraction f_M of corrupted parties and the fraction f_I of inactive users following our analysis. Protocol 1 is based on a standard signature scheme Sig, a key evolving signature scheme KES, a verifiable random function VRF and random oracles H, H'. In Appendix B, we analyze the protocol security.

Theorem 1. *Let VRF be a secure Verifiable Random Function,* KES *be a secure Key Evolving Signature scheme, Sig be a EUF-CMA secure signature scheme and H be a cryptographic hash function. Let $p = \theta/2^\ell$ and Q be the quorum guaranteeing a consistent view that at least one honest party has acted as defined in Section 3.3. Assuming a PKI and a random beacon, Protocol 1 is an Authenticated Verifiable Consistent Broadcast protocol (Definition 2) secure against Byzantine Adversaries over asynchronous networks.*

Realizing the Random Beacon. The random beacon required Protocol 1 can be realized without requiring extra assumptions from a VRF, via the resettable leaky beacon protocol of [22]. Although this protocol allows the adversary to introduce a bounded bias on the outputs, our novel analysis justifies choosing small quorum sizes even when there is such bounded bias in the beacon outputs. In Appendix B, we show how the beacon protocol of [22] can be executed in tandem with Protocol 1. Finally, we remark that aggregate forward secure VRFs [21] can be used to improve concrete efficiency by aggregating the VRF proofs and KES signatures.

Acknowledgements We thank Joakim Brorsson for insightful discussions and valuable feedback on this work. This work was partially supported by IRFD grant numbers 9040-00399B and 0165-00079B; VR grant numbers 2025-05869 and 2019-04166; SSF grant 2019-04166.

Protocol 1 – Protocol VCBC for party P_i and tag $ID.j.s$, parameterized by θ and Q, with access to a random beacon and a PKI registering all $(\mathsf{pk.Sig}_i, \mathsf{pk.KES}_i, \mathsf{pk.VRF}_i)$.

Initialization: Set $\bar{m} = \perp$, $\mathsf{ctr} = 0$, $\bar{W} = \perp$. Obtain r_s from the random beacon.

Upon receiving $(ID.j.s, \mathtt{in}, \mathsf{c} - \mathtt{broadcast}, m)$:

1: If $j = i$, proceed as follows:
2: Sign $\sigma_j \leftarrow \mathsf{Sig.Sign}(\mathsf{sk.Sig}_j, (ID.j.s, \mathsf{c} - \mathtt{send}, m))$.
3: Set $W \leftarrow \sigma_j$ and send $(ID.j.s, \mathsf{c} - \mathtt{send}, m, \sigma_j)$ to all parties.

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{send}, m, \sigma_j)$ from P_l:

1: **if** $l = j \wedge \bar{m} = \perp \wedge 1 = \mathsf{Sig.Verify}(\mathsf{pk.Sig}_j, (ID.j.s, \mathsf{c} - \mathtt{send}, m), \sigma_j)$
2: Set $\bar{m} \leftarrow m$
3: Compute $(y_i, \pi_i) \leftarrow \mathsf{VRF.Eval}_{\mathsf{sk.VRF}_i}(\mathsf{r}_s)$
4: **if** $y_i < \theta$
5: Compute $\sigma_i \leftarrow \mathsf{KES.Sign}(\mathsf{sk.KES}_i, (ID.j.s, \mathsf{c} - \mathtt{ready}, y_l, \pi_l, H(m)), s)$
6: Let $\mathsf{sk}'.\mathsf{KES}_i \leftarrow \mathsf{KES.KeyUpdate}(\mathsf{sk.KES}_i, s)$
7: Erase $\mathsf{sk.KES}_i$ and let $\mathsf{sk.KES}_i \leftarrow \mathsf{sk}'.\mathsf{KES}_i$
8: Send $(ID.j.s, \mathsf{c} - \mathtt{ready}, y_i, \pi_i, \sigma_i)$ to P_j.

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{ready}, y_l, \pi_l, \sigma_l)$ from P_l for the first time:

1: **if** $\begin{pmatrix} (i = j) \ \wedge \ (y_l < \theta) \ \wedge \ (1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_l, \mathsf{r}_s, y_l, \pi_l)) \\ \wedge\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_l, (ID.j.s, \mathsf{c} - \mathtt{ready}, y_l, \pi_l, H(m)), \sigma_l, s) \end{pmatrix}$
2: Set $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$ and $\bar{W} \leftarrow \bar{W} \cup \{(l, y_l, \pi_l, \sigma_l)\}$
3: **if** $\mathsf{ctr} > Q$
4: Send $(ID.j.s, \mathsf{c} - \mathtt{final}, \bar{W})$ to all parties.

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{final}, W)$

1: **if** $\begin{pmatrix} (|W| > Q) \ \wedge \ \forall (k, y_k, \pi_k, \sigma_k) \in W : \\ \begin{cases} y_l < \theta 1 \leftarrow \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathsf{r}_s, y_k, \pi_k) \\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_k, (ID.j.s, \mathsf{c} - \mathtt{ready}, y_k, \pi_k, H(\bar{m})), \sigma_k, s) \end{cases} \end{pmatrix}$
2: Set $\bar{W} \leftarrow W$ and output $(ID.j.s, \mathtt{out}, \mathsf{c} - \mathtt{deliver}, \bar{m})$

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{request})$ from P_l: (Verifiability)

1: **if** $\bar{W} \neq \perp$
2: Send $(ID.j.s, \mathsf{c} - \mathtt{answer}, \bar{m}, \bar{W}, \sigma_j)$ to P_l.

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{answer}, m, W, \sigma_j)$ from P_l: (Verifiability)

1: **if** $\begin{pmatrix} 1 = \mathsf{Sig.Verify}(\mathsf{pk.Sig}_j, (ID.j.s, \mathsf{c} - \mathtt{send}, m), \sigma_j) \\ \wedge\ (|W| > Q) \ \wedge \forall (k, y_k, \pi_k, \sigma_k) \in W : \\ \begin{cases} y_k < \theta \\ 1 \leftarrow \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathsf{r}_s, y_k, \pi_k) \\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_k, (ID.j.s, \mathsf{c} - \mathtt{ready}, y_k, \pi_k, H(\bar{m})), \sigma_k, s)) \end{cases} \end{pmatrix}$
2: Set $\bar{m} = m$ and $\bar{W} \leftarrow W$
3: Output $(ID.j.s, \mathtt{out}, \mathsf{c} - \mathtt{deliver}, \bar{m})$

References

1. Anonymous: Tail-hammer code. `https://github.com/werekorren/tail-hammer` (2025)

2. Backes, M., Berrang, P., Hanzlik, L., Pryvalov, I.: A framework for constructing single secret leader election from MPC. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds.) ESORICS 2022, Part II. LNCS, vol. 13555, pp. 672–691. Springer, Cham (Sep 2022). https://doi.org/10.1007/978-3-031-17146-8_33
3. Badertscher, C., Gazi, P., Kiayias, A., Russell, A., Zikas, V.: Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 913–930. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243848
4. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 431–448. Springer, Berlin, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1_28
5. Benhamouda, F., Gentry, C., Gorbunov, S., Halevi, S., Krawczyk, H., Lin, C., Rabin, T., Reyzin, L.: Can a public blockchain keep a secret? In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 260–290. Springer, Cham (Nov 2020). https://doi.org/10.1007/978-3-030-64375-1_10
6. Bienstock, A., Escudero, D., Polychroniadou, A.: On linear communication complexity for (maximally) fluid MPC. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 263–294. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_9
7. Blum, E., Leung, D., Loss, J., Katz, J., Rabin, T.: Analyzing the real-world security of the algorand blockchain. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) ACM CCS 2023. pp. 830–844. ACM Press (Nov 2023). https://doi.org/10.1145/3576915.3623167
8. Boneh, D., Eskandarian, S., Hanzlik, L., Greco, N.: Single secret leader election. In: AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020. pp. 12–24. ACM (2020). https://doi.org/10.1145/3419614.3423258, https://doi.org/10.1145/3419614.3423258
9. Boneh, D., Partap, A., Rotem, L.: Post-Quantum Single Secret Leader Election (SSLE) from Publicly Re-Randomizable Commitments. In: Bonneau, J., Weinberg, S.M. (eds.) 5th Conference on Advances in Financial Technologies (AFT 2023). Leibniz International Proceedings in Informatics (LIPIcs), vol. 282, pp. 26:1–26:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2023). https://doi.org/10.4230/LIPIcs.AFT.2023.26, https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.AFT.2023.26
10. Braun, L., Damgård, I., Orlandi, C.: Secure multiparty computation from threshold encryption based on class groups. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 613–645. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_20
11. Cachin, C., Kursawe, K., Petzold, F., Shoup, V.: Secure and efficient asynchronous broadcast protocols. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 524–541. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_31
12. Cascudo, I., David, B.: Publicly verifiable secret sharing over class groups and applications to DKG and YOSO. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part V. LNCS, vol. 14655, pp. 216–248. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58740-5_8
13. Cascudo, I., David, B., Garms, L., Konring, A.: YOLO YOSO: Fast and simple encryption and secret sharing in the YOSO model. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part I. LNCS, vol. 13791, pp. 651–680. Springer, Cham (Dec 2022). https://doi.org/10.1007/978-3-031-22963-3_22
14. Catalano, D., Fiore, D., Giunta, E.: Adaptively secure single secret leader election from DDH. In: Milani, A., Woelfel, P. (eds.) 41st ACM PODC. pp. 430–439. ACM (Jul 2022). https://doi.org/10.1145/3519270.3538424
15. Catalano, D., Fiore, D., Giunta, E.: Efficient and universally composable single secret leader election from pairings. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 471–499. Springer, Cham (May 2023). https://doi.org/10.1007/978-3-031-31368-4_17
16. Chen, J., Micali, S.: Algorand: A secure and efficient distributed ledger. Theor. Comput. Sci. **777**, 155–183 (2019). https://doi.org/10.1016/J.TCS.2019.02.001, https://doi.org/10.1016/j.tcs.2019.02.001
17. Choudhuri, A.R., Goel, A., Green, M., Jain, A., Kaptchuk, G.: Fluid MPC: Secure multiparty computation with dynamic participants. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 94–123. Springer, Cham, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84245-1_4
18. Daian, P., Pass, R., Shi, E.: Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 23–41. Springer, Cham (Feb 2019). https://doi.org/10.1007/978-3-030-32101-7_2
19. Damgård, I.B., Kamp, S.H., Loss, J., Nielsen, J.B.: Asynchronous YOSO a la paillier. Cryptology ePrint Archive, Paper 2025/128 (2025), https://eprint.iacr.org/2025/128
20. David, B., Deligios, G., Goel, A., Ishai, Y., Konring, A., Kushilevitz, E., Liu-Zhang, C.D., Narayanan, V.: Perfect MPC over layered graphs. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 360–392. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_12
21. David, B., Dowsley, R., Konring, A., Larangeira, M.: MUSEN: Aggregatable key-evolving verifiable random functions and applications. In: Galdi, C., Phan, D.H. (eds.) SCN 24, Part II. LNCS, vol. 14974, pp. 317–337. Springer, Cham (Sep 2024). https://doi.org/10.1007/978-3-031-71073-5_15

22. David, B., Gazi, P., Kiayias, A., Russell, A.: Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 66–98. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78375-8_3
23. David, B., Magri, B., Matt, C., Nielsen, J.B., Tschudi, D.: GearBox: Optimal-size shard committees by leveraging the safety-liveness dichotomy. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 683–696. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3559375
24. Deligios, G., Konring, A., Liu-Zhang, C.D., Narayanan, V.: Statistical layered MPC. In: Boyle, E., Mahmoody, M. (eds.) TCC 2024, Part IV. LNCS, vol. 15367, pp. 362–394. Springer, Cham (Dec 2024). https://doi.org/10.1007/978-3-031-78023-3_12
25. Gazi, P., Kiayias, A., Russell, A.: Fait accompli committee selection: Improving the size-security tradeoff of stake-based committees. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) ACM CCS 2023. pp. 845–858. ACM Press (Nov 2023). https://doi.org/10.1145/3576915.3623194
26. Gentry, C., Halevi, S., Krawczyk, H., Magri, B., Nielsen, J.B., Rabin, T., Yakoubov, S.: YOSO: You only speak once - secure MPC with stateless ephemeral roles. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 64–93. Springer, Cham, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84245-1_3
27. Gentry, C., Halevi, S., Magri, B., Nielsen, J.B., Yakoubov, S.: Random-index PIR and applications. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part III. LNCS, vol. 13044, pp. 32–61. Springer, Cham (Nov 2021). https://doi.org/10.1007/978-3-030-90456-2_2
28. Giunta, E., Stewart, A.: Unbiasable verifiable random functions. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part IV. LNCS, vol. 14654, pp. 142–167. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58737-5_6
29. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988). https://doi.org/10.1137/0217017, https://doi.org/10.1137/0217017
30. Hanke, T., Movahedi, M., Williams, D.: Dfinity technology overview series: Consensus system. https://dfinity.org/tech
31. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 357–388. Springer, Cham (Aug 2017). https://doi.org/10.1007/978-3-319-63688-7_12
32. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: 40th FOCS. pp. 120–130. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814584
33. Rachuri, R., Scholl, P.: Le mans: Dynamic and fluid MPC for dishonest majority. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part I. LNCS, vol. 13507, pp. 719–749. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15802-5_25
34. Zamani, M., Movahedi, M., Raykova, M.: RapidChain: Scaling blockchain via full sharding. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 931–948. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243853

A Building Blocks and Definitions

In this section we present more details on building block syntax and refer readers to the relevant references for full security definitions. Moreover, we recall the definition of consistent broadcast.

A.1 Cryptographic Building Blocks

Digital Signatures We make use of a standard EUF-CMA signature scheme Sig [29].

Definition 1 (Signature Scheme). *A signature scheme* Sig *with message space* $\mathcal{M}$ *consists of the following procedures:*

- $\mathsf{Setup}(\lambda) \to \mathsf{pp}$*: On input the security parameter* λ *output public parameters* pp*.*
- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{pk})$*: On input the public parameters* pp *output a secret/public key pair* $(\mathsf{sk}, \mathsf{pk})$*.*
- $\mathsf{Sign}(\mathsf{sk}, m) \to \sigma$*: On input a secret key* sk *and a message* $m \in \mathcal{M}$ *output a signature* σ*.*
- $\mathsf{Sig.Verify}(\mathsf{pk}, m, \sigma) \to 0/1$*: On input a public key* pk*, a message* $m \in \mathcal{M}$ *and signature* σ *output 0 (reject) or 1 (accept).*

Intuitively, a digital signature scheme is *correct* if honestly generated signatures are always accepted by the verifying algorithm, and *existentially unforgeable* if it is infeasible to generate a signature that is accepted by the verifying algorithm with non-negligible probability without the knowledge of the secret key. For formal definitions, see [29].

Key Evolving Signatures (KES) Key Evolving Signatures (KES) [4] are digital signature schemes in which the signing key can be evolved after every signature generation, such that it should be computationally infeasible to retrieve a previous signing key given the current (evolved) key, allowing for forward security with erasures. A KES consists of four algorithms:

- $\mathsf{KeyGen}(\lambda, \mathsf{Max}) \to (\mathsf{sk}, \mathsf{pk})$: on input a security parameter λ and a maximum number of signatures Max, outputs a secret key sk (interpreted as sk_0 for signing period $i = 0$) and a public key pk.
- $\mathsf{KeyUpdate}(\mathsf{sk}, i) \to \mathsf{sk}_{i+1}$: on input the secret key sk for signing period i, outputs the evolved secret key sk_{i+1} for signing period $i + 1$.
- $\mathsf{Sign}(\mathsf{sk}, m, i) \to \sigma_i$: on input a secret key sk for signing period $i \leq \mathsf{Max}$, a message m, and index i, outputs a signature σ_i.
- $\mathsf{Verify}(\mathsf{pk}, m, \sigma, i) \to 0/1$: on input a public key pk, a message m, a signature σ, and index i, outputs 1 iff σ_i is a valid signature on m under pk for i.

We require a KES *existentially unforgeable* under adaptive chosen message attacks (EUF-CMA). We also require that the KES has *forward security* (FSSIG), which intuitively means that an (adaptive) adversary cannot forge a signature for old keys, i.e., given the secret key of a user at the current time sk_i, it is computationally infeasible to generate signatures for previous keys sk_j with $j < i$. For formal definitions, see [4].

Verifiable Random Functions (VRF) Verifiable Random Functions (VRF) [32] allow a prover to compute a verifiably pseudorandom output from a given input. Committee selection applications require a VRF with *unpredictability under malicious key generation* (aka unbiasable), which prevents VRF outputs from being biased. A VRF consists of three algorithms:

- $\mathsf{KeyGen}(\lambda) \to (\mathsf{sk}, \mathsf{pk})$: on input a security parameter λ outputs a secret key sk and a public key pk.
- $\mathsf{Eval}(\mathsf{sk}, \mathsf{r}) \to (y, \pi_y)$: on input a secret key sk and a value r, outputs a pseudorandom value $y \in \{1, 2, \ldots, 2^\ell\}$ and a proof π_y.
- $\mathsf{Verify}(\mathsf{pk}, \mathsf{r}, y, \pi_y) \to 0/1$: on input a public key pk, a value r, a value y, and a proof π_y, outputs 1 if the proof verifies, and 0 otherwise.

A VRF is secure if it has *uniqueness*, *i.e.*, for each (pk, π_y) there is a single y which will verify; it has *provability*, *i.e.* a correctly produced y and π_y will always verify; and it has *pseudorandomness*, *i.e.* a computationally bounded adversary has a negligible advantage in distinguishing y from randomness. For formal definitions, see [32]. The above properties, however, do not account for malicious generation of VRF keys, which may impact the VRF output and skew the distribution of y. Informally, this guarantees that regardless of how the public key was generated, the output of the VRF must be random. For formal definitions and concrete constructions, see [22,28].

A.2 Consistent Broadcast

Consistent Broadcast [11] allows for a potentially corrupted sender to broadcast a message such that not all honest parties are guaranteed to output but all honest parties that do output are guaranteed output the same message. When the sender is honest, all honest parties are guaranteed to output the sender's message.

Definition 2 (Authenticated Verifiable Consistent Broadcast [11]). *Aprotocol for authenticated consistent broadcast satisfies the following conditions except with negligible probability:*

Validity: *If an honest party has* $\mathtt{c-broadcast}$ *m tagged with ID.j.s, then all honest parties* $\mathtt{c-deliver}$ *m tagged with ID.j.s, provided all honest parties have been activated on ID.j.s and the adversary delivers all associated messages.*

Consistency: *If some honest party* $\mathtt{c-delivers}$ *m tagged with ID.j.s and another honest party* $\mathtt{c-delivers}$ *m′ tagged with ID.j.s, then* $m = m'$.

Authenticity: *For all ID, senders j, and sequence numbers s, every honest party* $\mathtt{c-delivers}$ *at most one message m tagged with ID.j.s. Moreover, if* P_j *is honest, then m was previously* $\mathtt{c-broadcast}$ *by* P_j *with sequence number s.*

Efficiency: *For any ID , sender j, and sequence number s, the communication complexity of instance ID.j.s is uniformly bounded.*

Verifiability: *When an honest party has* $\mathtt{c-delivered}$ *m tagged with ID.j.s, then it can produce a single protocol message M that it may send to other parties such that any other honest party will* $\mathtt{c-deliver}$ *m tagged with ID.j.s upon receiving M (provided the other party has not already done so before).*

B Adaptively Secure Consistent Broadcast: Security Analysis and Beacon Realization

B.1 Security Analysis of Protocol 1

The security of Protocol 1 relies on obtaining VRF outputs $y_l > \theta$ (with respective proofs π_l) and KES signatures σ_l from Q-many parties. The quorum Q should guarantee a consistent view across all honest parties, as long as at least one honest party has provided y_l, π_l and that σ_l are KES signatures on the message m. Our analysis in Section 3.3 shows that such a quorum can be achieved when setting $p = \theta/2^\ell$ under different corruption ratios for the total set of parties.

The security properties of the VRF guarantee that the adversary cannot bias the committee election (or predict it), as proven in [22]. Moreover, since this VRF-based committee selection scheme keeps all parties anonymous until they act (as proven in [22]), an adaptive adversary cannot corrupt those parties before they answer with their message $(ID.j.s, \mathtt{c-ready}, y_l, \pi_l, \sigma_l)$. Since we obtain forward security with the help of the KES signature scheme, the adversary cannot withhold this message after it is sent and use keys obtained from a corrupted party to sign a conflicting message.

Given that Q valid values y_l, π_l, σ_l are present in W, we get the guarantee that at least one honest party has signed message m and that the adversary cannot obtain another set of Q such values signing a conflicting message $m' = m$. Hence, there can only exist sets W with Q signatures on one message m, which means that all honest parties that receive $(ID.j.s, \mathtt{c-final}, W)$ will output the same message m.

The verifiability property follows from the fact that both the signatures σ_l and the validity of the committee that generated them can be publicly verified. Notice that the committee is determined by the VRF outputs, which can be verified using their respective proofs. Moreover, as proven in [22], the adversary cannot forge such proof of committee election given the VRF security properties. Hence, every party who receives $(ID.j.s, \mathtt{c}-\mathtt{answer}, m, W)$ will correctly output m and the adversary cannot forge a message $(ID.j.s, \mathtt{c}-\mathtt{answer}, m', W')$ for $m' = m$.

B.2 Realizing the Randomness Beacon

In Protocol 2 and Protocol 3, we show how the beacon protocol of [22] can be executed in tandem with Protocol 1. For the most part, the protocol is executed exactly as Protocol 1 but extra steps (written in gray) are executed in order to generate randomness. The main idea follows the approach of [22], where each party in the committee elected for an execution with tag $ID.j.s$ helps generate randomness for the next execution with tag $ID.j.s$. The randomness is derived by evaluating the VRF on the current randomness r_0 concatenated with a string $\mathtt{RB}$ (for domain separation) to obtain an output and proof $(b_i, \pi_i') \leftarrow \mathsf{VRF.Eval}_{\mathsf{sk.VRF}_i}(\mathtt{RB}|\mathsf{r}_{s-1})$. The randomness for the next execution is then computed as $\mathsf{r}_{s+1} = H'(b_1, \ldots, b_k)$ for values $b_1, \ldots, b_k \in W$, where H' is a random oracle. Since the randomness for each execution can be determined from the VRF outputs, it can also be verified by verifying these outputs with their respective proofs and recomputing the value.

The security of the resulting protocol follows from the fact that the beacon protocol of [22] produces bounded bias random outputs and that Protocol 1 is an Authenticated Verifiable Consistent Broadcast protocol secure against Byzantine Adversaries over asynchronous networks. Notice that our quorum size analysis can deal with the beacon bias. When we consider a statistical security parameter of 128 (*i.e.* probability 2^{-128} of a quorum of a given size not satisfying the honesty conditions), our quorum sizes are still quite small and manageable. Hence, we can select such a high security parameter in order to withstand the degradation caused by the bounded bias the adversary may introduce in the beacon. In this case, a malicious party P_j could select which messages to include in W in order to further bias the beacon output beyond what is possible in the originals setting of [22]. However, notice that even if the adversary may grind over 2^64 different combinations of VRF outputs to choose which one to include in W, which is clearly infeasible, this would only degrade our security guarantees to an error probability of 2^{64}.

Protocol 2 – Protocol VCBC integrated with random beacon from [22] for party P_i and tag $ID.j.s$, parameterized by θ and Q, with access to a PKI registering all $(\mathsf{pk.Sig}_i, \mathsf{pk.KES}_i, \mathsf{pk.VRF}_i)$. Steps related to the random beacon are shown in gray color. Verifiability steps are shown in Protocol 3

Initialization: Set $\bar{m} = \perp$, $\mathsf{ctr} = 0$, $\bar{W} = \perp$. Obtain r_0 from common random string.

Upon receiving $(ID.j.s, \mathtt{in}, \mathsf{c} - \mathtt{broadcast}, m)$:

1: If $j = i$, proceed as follows:
2: Sign $\sigma_j \leftarrow \mathsf{Sig.Sign}(\mathsf{sk.Sig}_j, (ID.j.s, \mathsf{c} - \mathtt{send}, m))$.
3: Set $W \leftarrow \sigma_j$ and send $(ID.j.s, \mathsf{c} - \mathtt{send}, m, \sigma_j)$ to all parties.

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{send}, m, \sigma_j)$ from P_l:

1: **if** $l = j \wedge \bar{m} = \perp \wedge 1 = \mathsf{Sig.Verify}(\mathsf{pk.Sig}_j, (ID.j.s, \mathsf{c} - \mathtt{send}, m), \sigma_j)$
2: Set $\bar{m} \leftarrow m$
3: Compute $(y_i, \pi_i) \leftarrow \mathsf{VRF.Eval}_{\mathsf{sk.VRF}_i}(\mathsf{r}_s)$
4: **if** $y_i < \theta$
5: Compute $(b_i, \pi_i') \leftarrow \mathsf{VRF.Eval}_{\mathsf{sk.VRF}_i}(\mathtt{RB}|\mathsf{r}_{s-1})$
6: Compute $\sigma_i \leftarrow \mathsf{KES.Sign}(\mathsf{sk.KES}_i, (ID.j.s, \mathsf{c} - \mathtt{ready}, b_i, \pi_i', y_i, \pi_i, H(m)), s)$
7: Let $\mathsf{sk}'.\mathsf{KES}_i \leftarrow \mathsf{KES.KeyUpdate}(\mathsf{sk.KES}_i, s)$
8: Erase $\mathsf{sk.KES}_i$ and let $\mathsf{sk.KES}_i \leftarrow \mathsf{sk}'.\mathsf{KES}_i$
9: Send $(ID.j.s, \mathsf{c} - \mathtt{ready}, b_i, \pi_i', y_i, \pi_i, \sigma_i)$ to P_j.

Upon receiving the first $(ID.j.s, \mathsf{c} - \mathtt{ready}, b_l, \pi_l', y_l, \pi_l, \sigma_l)$ from P_l:

1: **if** $\left(\begin{array}{l} (i = j) \ \wedge \ (y_l < \theta) \\ \wedge\ 1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_l, \mathsf{r}_s, y_l, \pi_l) \\ \wedge\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_l, (ID.j.s, \mathsf{c} - \mathtt{ready}, \\ \qquad\qquad b_l, \pi_l', y_i, \pi_i, H(m)), \sigma_l, s) \\ \wedge\ 1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_l, \mathtt{RB}|\mathsf{r}_s, b_l, \pi_l) \end{array}\right)$
2: Set $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$ and $\bar{W} \leftarrow \bar{W} \cup (l, b_l, \pi_l', y_l, \pi_l, \sigma_l)$
3: **if** $\mathsf{ctr} > Q$
4: Set $\mathsf{r}_{s+1} = H'(\{b_l\}_{b_l \in \bar{W}})$
5: Send $(ID.j.s, \mathsf{c} - \mathtt{final}, \bar{W})$ to all parties

Upon receiving $(ID.j.s, \mathsf{c} - \mathtt{final}, W)$:

1: **if** $\left(\begin{array}{l} (|W| > Q) \ \wedge \ \forall\ (k, b_k, \pi_k', y_k, \pi_k, \sigma_k) \in W : \\ \left\{\begin{array}{l} y_k < \theta \\ 1 \leftarrow \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathsf{r}_s, y_k, \pi_k) \\ \wedge\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_k, (ID.j.s, \mathsf{c} - \mathtt{ready}, \\ \qquad\qquad b_k, \pi_k', y_k, \pi_k, H(\bar{m})), \sigma_k, s) \\ 1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathtt{RB}|\mathsf{r}_s, b_k, \pi_k) \end{array}\right. \end{array}\right)$
2: Set $\bar{W} \leftarrow W$ and $\mathsf{r}_{s+1} = H'(\{b_k\}_{b_k \in W})$
3: Output $(ID.j.s, \mathtt{out}, \mathsf{c} - \mathtt{deliver}, \bar{m})$

Protocol 3 – Verifiability for Protocol 2. Steps related to the random beacon are shown in gray color.

Upon receiving $(ID.j.s, \mathtt{c}-\mathtt{request})$ from P_l: (VERIFIABILITY)

1: **if** $\bar{W} \neq \perp$
2: Send $(ID.j.s, \mathtt{c}-\mathtt{answer}, \bar{m}, \bar{W}, \sigma_j)$ to P_l.

Upon receiving $(ID.j.s, \mathtt{c}-\mathtt{answer}, m, W, \sigma_j)$ from P_l: (VERIFIABILITY)

1: **if**
$$\left(\begin{array}{l} 1 = \mathsf{Sig.Verify}(\mathsf{pk.Sig}_j, (ID.j.s, \mathtt{c}-\mathtt{send}, m), \sigma_j) \\ \wedge\ (|W| > Q)\ \wedge\ \forall\ (k, b_k, \pi'_k, y_k, \pi_k, \sigma_k) \in W : \\ \left\{\begin{array}{l} y_k < \theta \\ 1 \leftarrow \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathsf{r}_s, y_k, \pi_k) \\ 1 \leftarrow \mathsf{KES.Verify}(\mathsf{pk.KES}_k, (ID.j.s, \mathtt{c}-\mathtt{ready}, \\ \qquad\qquad b_k, \pi'_k, y_k, \pi_k, H'(\bar{m})), \sigma_k, s)) \\ 1 = \mathsf{VRF.Verify}(\mathsf{pk.VRF}_k, \mathtt{RB}|\mathsf{r}_s, b_k, \pi_k) \end{array}\right. \end{array}\right)$$
2: Set $\bar{m} = m$, $\bar{W} \leftarrow W$ and $\mathsf{r}_{s+1} = H'(\{b_k\}_{b_k \in W})$
3: Output $(ID.j.s, \mathtt{out}, \mathtt{c}-\mathtt{deliver}, \bar{m})$

C Details on the formulas

Here we explain how we derived the formulas given in Section 4.3.

We begin with the case of quorums with at least one honest party, where the procedure to derive the formulas is the same for both asynchronous networks and reliable broadcast. The idea is to exploit the symmetry of the tails in the normal distribution:

$$\begin{aligned} \frac{Q - pX_1}{\sqrt{X_1 p(1-p)}} &= -\frac{Q - pX_2}{\sqrt{X_2 p(1-p)}} \\ \sqrt{X_2}(Q - pX_1) &= -(Q - pX_2)\sqrt{X_1} \\ \sqrt{X_2} \cdot Q + \sqrt{X_1} \cdot Q &= p(X_1\sqrt{X_2} + X_2\sqrt{X_1}) \\ Q(\sqrt{X_2} + \sqrt{X_1}) &= p\sqrt{X_2 X_1}(\sqrt{X_1} + \sqrt{X_2}) \\ Q &= p\sqrt{X_2 X_1} \end{aligned}$$

Next, we estimate the left tail using the *quantile* ϕ_b of the normal distribution (i.e. $\Pr(\mathcal{N} \leq \phi_b) = 2^{-b}$) and the equality we have just proven ($Q = p\sqrt{X_1 X_2}$) to derive the value of p:

$$\begin{aligned}
\frac{Q - pX_2}{\sqrt{X_2 p(1-p)}} &= \phi_b \\
Q - pX_2 &= \phi_b \sqrt{X_2 p(1-p)} \\
\left(p\sqrt{X_2 X_1} - pX_2\right)^2 &= \phi_b^2 X_2 p(1-p) \\
p^2 X_2 \left(\sqrt{X_1} - \sqrt{X_2}\right)^2 &= \phi_b^2 X_2 p(1-p) \\
p\left(\sqrt{X_1} - \sqrt{X_2}\right)^2 &= \phi_b^2 (1-p) \\
p\left[\left(\sqrt{X_1} - \sqrt{X_2}\right)^2 + \phi_b^2\right] &= \phi_b^2 \\
p &= \frac{\phi_b^2}{(\sqrt{X_2} - \sqrt{X_1})^2 + \phi_b^2}
\end{aligned}$$

The case of honest majority in a reliable broadcast follows the same idea as the at least one honest party case, using $\mathsf{f}_1(x) = \frac{x}{2}$ and $\mathsf{f}_2(x) = x$.

$$\begin{aligned}
\frac{\frac{Q}{2} - pX_1}{\sqrt{X_1 p(1-p)}} &= -\frac{Q - pX_2}{\sqrt{X_2 p(1-p)}} \\
\sqrt{X_2}\left(\frac{Q}{2} - pX_1\right) &= -(Q - pX_2)\sqrt{X_1} \\
\sqrt{X_2} \cdot \frac{Q}{2} + \sqrt{X_1} \cdot Q &= p(X_1\sqrt{X_2} + X_2\sqrt{X_1}) \\
Q\left(\frac{\sqrt{X_2} + 2\sqrt{X_1}}{2}\right) &= p\sqrt{X_2 X_1}(\sqrt{X_1} + \sqrt{X_2}) \\
Q &= 2p\sqrt{X_1 X_2}\frac{\sqrt{X_1} + \sqrt{X_2}}{2\sqrt{X_1} + \sqrt{X_2}}
\end{aligned}$$

Next, we estimate the left tail using the *quantile* ϕ_b of the normal distribution (i.e. $\Pr(\mathcal{N} \leq \phi_b) = 2^{-b}$) and the equality we have just proven ($Q = 2p\sqrt{X_1 X_2}\frac{\sqrt{X_1}+\sqrt{X_2}}{2\sqrt{X_1}+\sqrt{X_2}}$) to derive the value of p:

$$
\begin{aligned}
\frac{Q - pX_2}{\sqrt{X_2 p(1-p)}} &= \phi_b \\
Q - pX_2 &= \phi_b \sqrt{X_2 p(1-p)} \\
\left(2p\sqrt{X_1 X_2}\frac{\sqrt{X_1}+\sqrt{X_2}}{2\sqrt{X_1}+\sqrt{X_2}} - pX_2\right)^2 &= \phi_b^2 X_2 p(1-p) \\
p^2 X_2 \left(\frac{2X_1 + 2\sqrt{X_1X_2} - 2\sqrt{X_1X_2} - X_2}{2\sqrt{X_1}+\sqrt{X_2}}\right)^2 &= \phi_b^2 X_2 p(1-p) \\
p\left(\frac{2X_1 - X_2}{2\sqrt{X_1}+\sqrt{X_2}}\right)^2 &= \phi_b^2 (1-p)
\end{aligned}
$$

$$
\begin{aligned}
p\left(\left(\frac{2X_1 - X_2}{2\sqrt{X_1}+\sqrt{X_2}}\right)^2 + \phi_b^2\right) &= \phi_b^2 \\
p &= \frac{\phi_b^2}{\left(\frac{X_2 - 2X_1}{\sqrt{X_2}+2\sqrt{X_1}}\right)^2 + \phi_b^2}
\end{aligned}
$$

The case of honest majority in asynchronous networks is more complicated since it involves three bells. We derive a linear relation between p and Q from $\Pr(\mathcal{B}_{X_2} < \mathsf{f}_2(Q)) = \Pr(\mathcal{B}_{X_3} \geq \mathsf{f}_3(Q))$. Second, we express p as a function of the target quantile by imposing $\Pr(\mathcal{B}_{X_1} \geq \mathsf{f}_1(Q)) = 2^{-b}$. Third, we combine these two relations to compute Q. Finally, we verify that the resulting pair (p, Q) satisfies the three-bell safety-over-liveness condition, namely that

$$
\Pr(\mathcal{B}_{X_2} < \mathsf{f}_2(Q)) \quad = \quad \Pr(\mathcal{B}_{X_3} \geq \mathsf{f}_3(Q)) \quad < \quad \Pr(\mathcal{B}_{X_1} \geq \mathsf{f}_1(Q)) \quad = \quad 2^{-b}.
$$

If the check fails, we increase p to $p = \frac{pX_1+1}{X_1}$, recompute Q, and repeat the verification.

In details, first we exploit the symmetry of the tails for $\mathcal{B}_{X_2}$ and $\mathcal{B}_{X_3}$:

$$
\begin{aligned}
\frac{Q - pX_3}{\sqrt{X_3 p(1-p)}} &= -\frac{Q - pX_2}{\sqrt{X_2 p(1-p)}} \\
\sqrt{X_2}(Q - pX_3) &= -(Q - pX_2)\sqrt{X_3} \\
\sqrt{X_2}\cdot Q + \sqrt{X_3}\cdot Q &= p(X_3\sqrt{X_2} + X_2\sqrt{X_3}) \\
Q(\sqrt{X_2}+\sqrt{X_3}) &= p\sqrt{X_2X_3}(\sqrt{X_3}+\sqrt{X_2}) \\
Q &= p\sqrt{X_2X_3}
\end{aligned}
$$

Next we estimate the left tail using the right *quantile* ϕ_b of the normal distribution (i.e. $\Pr(\mathcal{N} \geq \phi_b) = 2^{-b}$) and the equality we have just proven ($Q = p\sqrt{X_3X_2}$) to derive the value of p:

$$
\begin{aligned}
\Pr\left(\mathcal{N} \geq \frac{\mathsf{f}_1(Q) - pX_1}{\sqrt{X_1 p(1-p)}}\right) &= 2^{-b} \\
\frac{\frac{Q}{2} - pX_1}{\sqrt{X_1 p(1-p)}} &= \phi_b \\
\frac{Q}{2} - pX_1 &= \phi_b \sqrt{X_1 p(1-p)} \\
\frac{p\sqrt{X_3 X_2}}{2} - pX_1 &= \phi_b \sqrt{X_1 p(1-p)} \\
p^2 X_1 \left(\frac{\sqrt{X_3 X_2}}{2\sqrt{X_1}} - \sqrt{X_1}\right)^2 &= \phi_b^2 X_1 p(1-p) \\
p \left(\frac{\sqrt{X_3 X_2}}{2\sqrt{X_1}} - \sqrt{X_1}\right)^2 &= \phi_b^2 (1-p) \\
p &= \frac{\phi_b^2}{\left(\frac{\sqrt{X_2 X_3} - 2X_1}{\sqrt{4X_1}}\right)^2 + \phi_b^2}
\end{aligned}
$$

D Considerations on the Closed Formulas for p and Q

From the formulas in Section 4.3, we observe that both the probability and the quorum depend solely on the configuration of the binomial distributions X_2 and X_1 and the required security level b. The probability p always takes a value between 0 and 1, while the quorum Q is proportional to this probability (specifically, the probability multiplied by a constant). Furthermore, we find that the quorum Q is smaller than the expected committee size.

The limitations of these formulas arise from the error introduced by approximating the binomial distribution with the normal distribution. The normal distribution provides a better approximation as the size of the binomial population increases and the success probability approaches 0.5. Since in our case the probability is significantly smaller than 0.5, the formula must be adjusted in order to obtain a better fit between the normal and the binomial distributions. We denote by p and Q the probability and the quorum obtained from the formulas in 4.3, whereas $\widetilde{p}$ and $\widetilde{Q}$ represent the actual values employed in the protocol to ensure the required security guarantees. In order to refine the approximation and obtain the desired result, we may apply two adjustments. First, we set the security level slightly higher than the required threshold. Second, we observe that the quorum $\widetilde{Q}$ for the binomial distribution lies within a close range of the quorum Q derived from the normal approximation. Hence, from the quantile $\Pr(\mathcal{N} \leq \widetilde{\phi}) = \widetilde{\epsilon} < \epsilon$, we can compute the probability $\widetilde{p}$ exactly as p through the formulas in 4.3 and then the quorum $\widetilde{Q}$ will lie within the range of Q given by the formula, $\widetilde{Q} = Q \pm \Delta$. A rule of thumb derived from heuristics is to set $\widetilde{\epsilon} \sim 2^{-0.2}\epsilon$ and $\Delta \in [0, 100]$. This implies that the resulting formulas can estimate the quorum size $\widetilde{Q}$ and the probability $\widetilde{p}$ and the error as relatively small compared to the size of the population. These formulas are particularly useful when N is 10^4 or larger. For smaller populations, it is generally more practical to compute the probabilities by enumerating all possible cases, since the population is manageable and the error introduced by approximating the binomial distribution with the normal distribution becomes more significant.

Approximating the binomial distribution with the normal distribution provides significant advantages: for instance, the normal distribution possesses symmetry properties, and its quantiles are more efficiently implemented than those of the binomial distribution. For example, in R, there exists a library for computing the quantiles of the normal distribution using the Multiple Precision Floating-Point Reliable (MPFR) arithmetic library.