$\log^*$-Round Game-Theoretically-Fair Leader Election

Ilan Komargodski[*], Shin'ichiro Matsuo[†], Elaine Shi[‡], Ke Wu[§]

Abstract

It is well-known that in the presence of majority coalitions, *strongly fair* coin toss is impossible. A line of recent works have shown that by relaxing the fairness notion to game theoretic, we can overcome this classical lower bound. In particular, Chung et al. (CRYPTO'21) showed how to achieve approximately (game-theoretically) fair leader election in the presence of majority coalitions, with round complexity as small as $O(\log \log n)$ rounds.

In this paper, we revisit the round complexity of game-theoretically fair leader election. We construct $O(\log^* n)$ rounds leader election protocols that achieve $(1-o(1))$-approximate fairness in the presence of $(1-O(1))n$-sized coalitions. Our protocols achieve the same round-fairness trade-offs as Chung et al.'s and have the advantage of being conceptually simpler. Finally, we also obtain game-theoretically fair protocols for committee election which might be of independent interest.

[*] Department of Computer Science, Hebrew University of Jerusalem and NTT Research. Email: ilank@cs.huji.ac.il.

[†] NTT Research and Department of Computer Science, Georgetown University. Email: shinichiro.matsuo@ntt-research.com

[‡] Computer Science Department, Carnegie Mellon University. Email: runting@gmail.com

[§] Computer Science Department, Carnegie Mellon University. Email: kew2@andrew.cmu.edu

Contents

1 Introduction 2
1.1 Our Results and Contributions . . . 3
1.2 Additional Related Work . . . 5

2 Technical Roadmap 6
2.1 Electing Poly-logarithmically Sized Committees: Achieving CSP-Fairness . . . 6
2.2 Electing Poly-logarithmically Sized Committees: Achieving Maximin Fairness . . . 7
2.3 Leader Election . . . 8

3 Preliminaries 9
3.1 Probability Tools . . . 9
3.2 Fairness Notions for Committee Election . . . 9
3.3 Publicly Verifiable Concurrent Non-Malleable Commitment . . . 12

4 Game-Theoretically Fair Committee Election 13
4.1 Electing Poly-logarithmically Sized Committees: Achieving CSP-Fairness . . . 14
4.2 Electing Poly-logarithmically Sized Committees: Achieving Maximin-Fairness . . . 18
4.2.1 Anonymous Broadcast Functionality . . . 18
4.2.2 Formal Description of the Protocol . . . 19
4.2.3 Proof of Lemma 4.8 . . . 20
4.2.4 Proof of Lemma 4.9 . . . 21
4.2.5 Proof of Lemma 4.10 . . . 23

5 Fairness Amplification Though Iteration 23
5.1 Preliminary: Fairness of Tournament Tree Protocol . . . 24
5.2 Our Final Game-Theoretically Fair Committee Election . . . 24

6 Instantiation of the Ideal Functionalities 31
6.1 Ideal Zero-Knowledge Functionality $\mathsf{IdealZK}^*$ . . . 31
6.2 Implementing Anonymous Broadcast Functionality . . . 32

7 Open Questions 39

A Proof of Theorem 6.1 42
A.1 Public verifiability . . . 42
A.1.1 Bounded-concurrent public-coin zero-knowledge proof . . . 42
A.1.2 Public verifiable commit-with-extract . . . 42
A.2 Realizing the $\mathsf{MemberZK}^*$-hybrid model . . . 45
A.3 Reduce $\mathsf{IdealZK}^*$ to $\mathsf{MemberZK}^*$ . . . 48

1 Introduction

Suppose that Murphy, Murky, and Moody co-authored a paper that proved a ground-breaking theorem and the paper got accepted at the prestigious CRYPTO'22 conference. Murphy, Murky, and Moody want to run a coin toss protocol over the Internet to elect a winner who will present the paper at the conference. Since everyone wants to go to the beautiful beaches of Santa Barbara, all of them want to be the winner. They each are worried that the other coauthors might deviate from the honest protocol to gain an unfair advantage. There is both good and bad news. The bad news is that due to a famous lower bound by Cleve [Cle86], there is no *strongly fair* coin toss protocol when half of the parties may be corrupt and misbehaving — roughly speaking, strong fairness requires that the coalition cannot bias the outcome of the coin toss whatsoever. The good news is that a more recent line of work [CCWS21, GGS, CGL+18, WAS22] has shown that a relaxed fairness notion called *game-theoretic* fairness is indeed possible for the leader election problem, even when an arbitrary number of parties may be corrupt. To see why, first observe that the original Blum's coin toss protocol [Blu83] actually gives a game-theoretically fair leader election scheme for $n = 2$ parties. Imagine that each party first commits to a random coin, they then open their coin, and the XOR of the two bits is used to elect a random winner. If one party fails to commit or correctly open, it is eliminated and the remaining party is declared the winner. Blum's coin toss satisfies *game-theoretic* fairness in the following sense. As long as the commitment scheme is not broken, a corrupt layer cannot bias the coin *to its own favor* no matter how it deviates from the protocol. Note that Blum's protocol is not strongly fair since a corrupt party can indeed bias the coin, but only to the other player's advantage.

For the more general case of the n parties, we can use a folklore tournament-tree protocol to accomplish the same purpose. Suppose that n is a power of 2 for simplicity. We first divide the n parties into $n/2$ pairs, and each pair elects a winner using Blum's coin toss. The winner survives to the next round, where we again divide the surviving $n/2$ parties into $n/4$ pairs. The protocol continues after a final winner is elected after $\log_2 n$ rounds. At any point in the protocol, if a party fails to commit or correctly open its commitment, it is eliminated and its opponent survives to the next round.

The recent work of Chung et al. [CCWS21] argued that this simple tournament tree protocol satsfies a strong notion of game-theoretic fairness as explained below. Suppose that the winner obtains a utility of 1 and everyone else obtains a utility of 0. As long as the commitment scheme is not broken, the tournament tree protocol guarantees that 1) no coalition of any size can *increase its own expected utilty* no matter what (polynomially-bounded) strategy it adopts; and 2) no coalition of any size can *harm any individual honest player's expected utility*, no matter what (polynomially-bounded) strategy it adopts. Recent work in this space [CCWS21, GGS, CGL+18, WAS22] calls the former notion cooperative-strategy-proofness (or *CSP-fairness* for short), and calls the latter notion *maximin fairness*. Philosophically, CSP-fairness guarantees that any rational, profit-seeking individual or coalition has no incentive to deviate from the honest protocol; and maximin fairness ensures that any paranoid individual who wants to maximally protect itself in the worst-case scenario has no incentive to deviate either. In summary, the honest protocol is an equilibrium and also the best response for every player and coalition. Therefore, prior works [CGL+18, CCWS21, WAS22, GGS] have argued that game-theoretic notions of fairness are compelling and worth investigating because 1) they are arguably more natural (albeit stricly weaker) than the classical strong fairness notion in practical applications; and 2) the game-theoretic relaxation allows us to circumvent classical impossibility results pertaining to strong fairness in the presence of majority coalitions [Cle86].

Having established the general feasibility of game-theoretically fair leader election in the presence of majority-sized coalitions, Chung et al. [CCWS21] asked the following natural question: *what is the round complexity of game-theoretically fair leader election in the presence of majority coalitions*? Specifically, can we asymptotically outperform the logarithmic round complexity of the folklore tournament tree protocol? They then gave a partial answer to this question, showing that for any desired round complexity parameter $\Theta(\log\log n) \leq R \leq \log n$, there is an $O(R)$-round n-party leader election protocol that achieves $\left(1 - \frac{1}{2^{\Theta(R)}}\right)$-fairness against coalitions of size up to $\left(1 - \frac{1}{2^{\Theta(R)}}\right) n$. In particular, their result statement adopts an approximate notion of game-theoretic fairness. Roughly speaking, a protocol is $(1-\epsilon)$-fair if it satisfies the aforementioned game theoretic fairness (including CSP-fairness and maximin fairness) up to an ϵ slack. More specifically, we want that the coalition's expected utility cannot exceed $1/(1-\epsilon)$ times its normal utility had everyone behaved honestly, and we require that any honest individual's expected utility cannot drop below $(1-\epsilon)$ times its normal utility had everyone behaved honestly. Chung et al.'s result [CCWS21] enables a smooth and mathematically quantifiable tradeoff between the efficiency of the protocol and its resilience to strategic behavior. However, their result requires the protocol to have at least $\Theta(\log\log n)$ rounds to give any meaningful fairness guarantee. Indeed, a more careful examination suggests that their framework has a *sharp* cutoff at $\Theta(\log\log n)$ rounds, i.e., the approach fundamentally fails when we want round complexity to be less than $\log\log n$. Therefore, an obvious gap in our understanding is the following:

> *In the presence of majority-sized coalitions, can we achieve any meaningful fairness guarantee for small-round protocols whose round complexity is less than* $\log\log n$?

1.1 Our Results and Contributions

In this paper, we revisit the round complexity of game-theoretically fair leader election. We make the following contributions. First, we show positive results in the style of Chung et al. [CCWS21], but now for a broader range of parameters as explained in the following Theorem 1.1. In particular, our result shows that under standard cryptographic assumptions, there is a $O(\log^* n)$-round leader election protocol that achieves $(1-o(1))$-game-theoretic-fairness, in the presence of $(1-O(1))\cdot n$-sized coalitions.

Second, we give conceptually simpler constructions than those of Chung et al. [CCWS21], which also result in simpler analyses. More specifically, Chung et al.'s construction relies on combinatorial objects called extractors, which we get rid of in our construction. We believe that our conceptually simpler constructions can lend to better understanding and make it easier for future work to extend our framework. Interestingly, our constructions are inspired and have structural resemblance to Feige's famous lightest bin leader election protocol [Fei99]. We stress, however, that Feige's protocol itself does not satisfy game-theoretic fairness, but rather, achieves only a much weaker notion of resilience, i.e., an honest party is elected leader with constant probability. At a very high level, our approach augments Feige's protocol lightest-bin protocol with a "commit and open" and a "virtual identity" mechanism, and we prove that the resulting protocol satisfies the desired game-theoretic properties.

Third, we also present results for the more generalized problem of fair committee election, where the goal is to elect a committee of size c. The leader election problem can be viewed as a special case of committee election where $c = 1$. Our main results are summarized in the following theorems.

Theorem 1.1 (Game-theoretically fair leader election)**.** *Assume the existence of enhanced trapdoor permutations, and collision-resistant hash functions. Fix* n *and let* $\log^* n \leq R \leq C \log n$ *be the round complexity we want to achieve for some constant* C*. Then there exists an* $O(R)$*-round leader* *election that achieves* $(1 - \frac{1}{2^{\Theta(R)}})$*-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most* $(1 - \frac{L}{\Theta(R)})n$*, where* L *is the smallest integer such that* $\log^{(L)} n \leq 2^R$*.*

For readers who are familiar with the line of work on approximate *strong* fairness [Cle86, MNS09, AO16, BOO10, HT14], an interesting observation is that for game-theoretic fairness, the efficiency-fairness tradeoff is exponentially better than that of strong fairness. Specifically, it is known that any R-round protocol cannot achieve $\Omega(1/R)$ *strong* fairness[1] against an $n/2$-sized coalition, whereas we show that R-round protocols can achieve $(1 - 1/2^{\Theta(R)})$-fairness.

Theorem 1.2 (Game-theoretically fair committee election)**.** *Assume the existence of enhanced trapdoor permutations and collision-resistant hash functions. Fix* n *and* c*. Let* L^* *be the smallest integer such that* $\log^{(L^*)} n \leq c$*. Then for any* $L^* \leq R \leq C_0 \log n$ *for some constant* C_0*, we have that*

- *If* $c \geq 2^R$*, there exists an* $O(R)$*-round committee election that achieves* $(1 - \frac{1}{c^{\Theta(1)}})$*-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most* $(1 - \frac{L^*}{\Theta(R)})n$*.*
- *If* $c < 2^R$*, there exists an* $O(R)$*-round committee election that achieves* $(1 - \frac{1}{2^{\Theta(R)}})$*-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most* $(1 - \frac{L}{\Theta(R)})n$*, where* L *is the smallest integer such that* $\log^{(L)} n \leq 2^R$*.*

Below are some interesting examples with respect to different committee size c and the round complexity R.

- For committee size $c = 1$, i.e., leader election, and round complexity $R = O(\log^* n)$, our protocol achieves $\Theta(1)$-game-theoretic fairness against a coalition of size $\Theta(n)$ assuming $\log *n$ is a constant;
- For committee size $c = 1$, i.e., leader election, and round complexity $R = \log \log \log n$, out protocol achieves $(1 - \frac{1}{\mathsf{poly} \log \log n})$-fairness against a coalition of size $n - \frac{n}{\Theta(\log \log \log n)}$.
- For committee size $c = \mathsf{poly} \log \log n$ and for constant round complexity $R = \Theta(1)$, our protocol achieves $(1 - \frac{1}{\mathsf{poly} \log \log n})$-fairness against $\Theta(n)$-sized coalition.

In this paper, we consider the standard notions of approximate CSP-fairness and maximin-fairness. The standard notion of approximate CSP-fairness is also sometimes referred to as *approximate coalition-resistant Nash equilibrium* in some earlier works such as Fruitchain [PS17]. It is also known [CCWS21] that the standard notion of approximate CSP-fairness (or maximin-fairness) is equivalent in some sense to approximate notions of fairness formulated by the more classical Rational Protocol Design (RPD) paradigm [GKM+13, GTZ15, GKTZ15].

Although the standard notion of approximate fairness seems the most natural one, Chung et al. [CCWS21] pointed out that when defining approximate fairness, one can in fact adopt a strengthened notion which they call sequential fairness. Their game-theoretically fair leader election result is in fact stated for the sequential notion. In this sense, our result is incomparable to theirs: they consider a stronger solution concept but their approach inherently cannot give any meaningful result for protocols of $o(\log \log n)$ rounds. By contrast, we consider the more standard non-sequential notion and we are able to generalize the smooth tradeoff between efficiency and fairness shown by Chung et al. [CCWS21] to a broader range of parameters.

[1] The approximate strong fairness line of work defines what we call $(1 - \epsilon)$-fairness as ϵ-fairness (but for the notion of strong fairness instead). Following the notations of Chung et al. [CCWS21], we flipped this notation to make it more intuitive: with our notation, 1-fair is more fair than 0-fair which agrees with our intuition.

1.2 Additional Related Work

Game theory meets cryptography. Some recent efforts have instigated the intersection of the game theory [Nas51, Aum74] and multi-party computation [GMW19, Yao82]. See [Kat08, DR+07] for a survey. There have been two classes of questions that have attracted a lot of interests.

Some work [HT04,KN08,ADGH06,OPRV09,AL11,ACH11] explore how to define game-theoretic notions of security, as opposed to cryptography security notions for distributed computing tasks such as secure function evaluation. Existing works in this line considered a different notion of utility than our work. Their utility functions are often defined assuming that players prefer to compute the function correctly, or prefer to learn others' secret data and prefers that other players do not gain knowledge about their own secrets. Garay et al. propose a paradigm called Rational Protocol Design [GKM+13] and develop this paradigm in subsequent works [GTZ15, GKTZ15]. As mentioned in Section 1, the standard notion of approximate CSP-fairness (or maximin fairness) is in some sense equivalent to the approximate notion of fairness formulated in RPD paradigm.

Another line of work explores how cryptography can help traditional game theory. Many works in game theory assumed the existence of a trusted mediator, which can be realized under cryptography [DHR00, IML05, GK12, BGKO11].

Recently, there has been renewed interest in the connection between game theory and cryptography. Besides the work of Chung et al. [CCWS21] that inspires our work, and [GGS] that generalized the lower bound of the round complexity of game-theoretically fair leader election, the recent work [CGL+18, WAS22] have also suggested game-theoretically fair multi-party binary-coin toss. Binary-coin toss considers tossing a binary coin among n players, while in leader election, we consider tossing an n-way coin among n players. These two formulations are different and they exhibit starkly different theoretical landscape.

Leader election in other models. Leader election has been studied extensively. A line of work [BK14, ADMM14] considered how to achieve "financially-fair" n-party lottery over cryptocurrencies. Their game-theoretic notion of fairness is similar to ours, yet they rely on collateral and penalty mechanisms to achieve fairness. As a comparison, our fairness can be achieved without relying on additional assumptions such as collateral and penalty. Moreover, [ADGH06] studied an incomparable game-theoretic notion for leader election. In their notions, all users prefer to have a leader, and users may have different preferences of who the leader is.

Besides, leader election was considered in the full information model [RZ01, RSZ02, Fei99, Dod06]. Their notion of security concentrates on electing an honest leader with some *small constant* probability, assuming honest majority [Fei99]. This notion is much weaker than the game-theoretic notion considered in our work, which are more suitable in some decentralized applications, where honest majority assumption is not applicable. Moreover, in the full-information model, leader election is impossible against a majority coalition even under this weak notion of security. Interestingly, our committee election protocol actually builds on Feige's lightest bin protocol [Fei99].

Approximate strong fairness. As mentioned in Section 1, the *de facto* notion of fairness considered in the multi-party computation literature is strong fairness or unbiasability. The celebrated result of Cleve [Cle86] showed that it is not possible to achieve $\Omega(\frac{1}{R})$-unbiasable coin toss against a coalition consisting of half or more players. Moran et al. [MNS09] showed how to obtain an R-round protocol that achieves $\Omega(\frac{1}{R})$-unbiasability in the two-party setting, that matches Cleve's lower bound. Recent work [AO16,BOO10,HT14] have been making encouraging progress on building fair multi-party coin toss. However, they rely on constant number of players to ensure polynomial round complexity. We cannot directly rely on multi-party unbiasable coin toss to build game-theoretically fair leader election because our trade-off curve between round complexity and the fairness slack ϵ is exponentially better than that of the unbiasability.

2 Technical Roadmap

2.1 Electing Poly-logarithmically Sized Committees: Achieving CSP-Fairness

We start by observing that a single iteration of Feige's lightest-bin protocol [Fei99] can elect a committee of size $c \geq \mathsf{poly}\log n$ while satisfying *CSP-fairness against relatively large coalitions.* Feige's ingenious protocol works as follows (we describe a single iteration of the protocol): each player $i \in [n]$ chooses a random bin b_i among a total of $B = n/c$ bins, and broadcasts its choice b_i. At this moment, we identify the lightest bin, and everyone who has placed itself in the lightest bin is elected as a committee member. A simple analysis shows that this protocol satisfies CSP-fairness against relatively large coalitions. Specifically, the lightest bin cannot exceed a capacity of $c = n/B$. Moreover, applying the standard Chernoff bound and the union bound, we know that with probability at least $1 - n \cdot \exp(-\Omega(\epsilon^4 \cdot c))$, a good event that every bin has at least $(1-\epsilon^2) \cdot (1-\beta) \cdot c$ honest players must happen, where $\beta \cdot n$ is the maximum coalition size for $\beta \in (0,1)$. Now we show that if the coalition has size larger than $\epsilon \cdot n$, then Feige's lightest bin is $(1-\Theta(\epsilon))$-CSP-fair. Given that the good event happens, the expected fraction of corrupted players in the committee is at most $1 - (1-\epsilon^2) \cdot (1-\beta) \leq \frac{\beta}{1-2\epsilon}$. For large n, it is easy to see that the good event happens with $1 - \mathsf{negl}(n)$ probability and the expected fraction of coalition in the committee is at most $\frac{\beta}{1-\Theta(\epsilon)}$. For small n, however, the calculation is more involved, as we will describe below. The overall expected fraction of the coalition in the committee is at most $\frac{\beta}{1-2\epsilon} + \delta$, where $\delta = n \cdot \exp(-\Omega(\epsilon^4 \cdot c))$ is the probability that the good event does not happen. To guarantee that the expected fraction of the coalition in the committee is at most $\frac{\beta}{1-\Theta(\epsilon)}$, we need the failure probability $\delta \leq \beta \cdot \Theta(\epsilon)$. The expected fraction of the coalition in the committee is thus $\frac{\beta}{1-2\epsilon} + \delta \leq \beta(\frac{1}{1-2\epsilon} + \Theta(\epsilon)) \leq \frac{\beta}{1-\Theta(\epsilon)}$. For example, if we pick $\epsilon = \frac{1}{\log n}$ and $c = (\log n)^{10}$, then the probability that the good event does not happen is at most $n \exp\{-\Omega((\log n)^6)\} \leq \epsilon^2 \leq \beta \cdot \epsilon$ for any $n \geq 3$. Henceforth the protocol satisfies $(1-\Theta(\epsilon))$ -CSP-fairness as long as the coalition contains at least ϵn players.

Unfortunately, the protocol does not satisfy CSP-fairness for small coalitions. For example, a single individual $i \in [n]$ (i.e., a coalition of size 1) can examine all others' bin choices and then decide to place itself in the lightest bin. In this case, if the lightest bin (not counting player i) is at least 2 lighter than the second lightest bin, player i is elected into the committee. This happens with a probability at least $\frac{6}{5} \cdot \frac{c}{n}$ for large n , which is significantly higher than the normal probability c/n that player i ought to be elected in an all-honest execution.

Commit-and-reveal lightest bin. We introduce commit-and-reveal version of Feige's lightest bin protocol which achieves CSP-fairness not just against large coalitions, but also against small coalitions as well. The idea is quite simple — below we describe the scheme assuming ideal commitments, although in our formal technical sections we will instantiate the commitments using standard non-malleable commitments. Everyone first commits to a random bin number among $B = n/c$ bins. They then open their commitments. Those who land in the lightest bin are declared the committee, and like before, anyone who fails to commit or correctly open is kicked out. Using the same argument as before, we can show that the commit-and-reveal lightest bin protocol also achieves $(1-\Theta(\epsilon))$-CSP-fairness against coalitions of size at least ϵn .

We now argue why it also satifies CSP-fairness against small coalitions of size $\beta n < \epsilon n$. Intuitively, the coalition's best strategy is to pick a bin with the fewest number of honest players (henceforth called the *honest-lightest* bin), and place as many coalition members in it as possible while still maintaining that it is the lightest. However, the coalition does not know which one is the honest-lightest bin when committing to its own bin choices. In fact, even when conditioned on the coalition's view during the commitment phase, each bin is the honest-lightest bin with equal probability. No matter how the coalition spreads its members across the bins, the expected number of coalition members in a *randomly chosen* bin is at most $\beta \cdot n/B = \beta \cdot c$. Further, with $1 - n \cdot \exp(-\Omega(\epsilon^4 \cdot c))$ probability, the good event that honest-lightest bin should have at least $(1-\epsilon^2)(1-\beta)c$ honest players happens. Therefore, the coalition's expected representation on the committee cannot exceed $\frac{\beta}{(1-\epsilon^2)(1-\beta)} \leq \frac{\beta}{1-2\epsilon}$ given that the good event happens. Overall, the expected fraction of the coalition in the committee is at most $\frac{\beta}{1-2\epsilon} + \delta$, where $\delta = n \cdot \exp(-\Omega(\epsilon^4 \cdot c))$ is the probability that the good event does not happen. Still, as long as $\delta \leq \beta\epsilon$, by the same analysis as before, the expected fraction of the coalition in the committee is at most $\frac{\beta}{1-\Theta(\epsilon)}$.

2.2 Electing Poly-logarithmically Sized Committees: Achieving Maximin Fairness

Although simple and cute, the commit-and-reveal lightest bin protocol does not satisfy maximin fairness. For example, a $\Theta(n)$-sized coalition can target a victim player $i \in [n]$ and prevent it from being elected with high probability using the following strategy. During the commitment phase, spread the coalition members evenly across all bins. During opening, first observe which bin (denoted b^*) player i lands in. Then, all coalition members fail to open except those whose choice was b^*.

To achieve maximin fairness, we are inspired by a virtual identity technique originally proposed by Chung et al. [CCWS21], but unfortunately, directly applying this idea to the lightest bin does not work. At a high level, a strawman idea is as follows:

1. Every player $i \in [n]$ selects a random virtual identity v_i from a sufficiently large space, and commits to the pair (i, v_i).
2. Every player $i \in [n]$ selects a random bin b_i among $B = n/c$ bins, and commits to the pair (v_i, b_i) where v_i is its secret virtual identity.
3. Everyone $i \in [n]$ opens their commitment of (v_i, b_i). The virtual identities contained in the lightest bin will be elected committee.
4. Everyone opens their real-virtual identity mapping (i, v_i). This will allow everyone to compute the real identities of those elected to the committee.

Now, as long as the coalition does not know an honest player i's virtual ID, it does not know who to target during the commit-and-reveal lightest bin steps (Steps 2 and 3). Therefore, as long as the good event that each bin contains at least $(1-\epsilon)(1-\beta)c$ honest players happens, an honest player i will be elected into the committee with probability at least $\frac{(1-\epsilon)(1-\beta)c}{(1-\beta)n} = \frac{(1-\epsilon)c}{n}$. By law of total probability, the probability that an honest player i gets elected into the committee with probability at least $\frac{(1-\epsilon)(1-\delta)c}{n}$, where $1-\delta$ is the probability that the good event happens. Henceforth, as long as $\delta \leq \epsilon$, an honest player i gets elected into the committee with probability at least $\frac{(1-\Theta(\epsilon))c}{n}$.

Unfortunately, this idea does not work if the coalition can eavesdrop on the network channel and observe who sent which (bin, virtual ID) pair in the commit-and-reveal lightest bin protocol.

This would allow the coalition to immediately learn the correspondance between virtual and real identities.

To salvage this idea, our high-level idea is simple but realizing it turns out to be somewhat subtle as we explain later. First, if we are willing to assume the existence of an idealized anonymous communication network where players can post messages anonymously, then we can overcome the aforementioned problem by running Steps 2 and 3 over an anonymous communication network. Therefore, it suffices to find a suitable anonymous communication protocol to realize anonymous communication. Although anonymous communication has been extensively studied in the literature [Cha81, Cha88, Abe99, CGF10, DMS04, SGR99, ZZZR05], in our setting, it is tricky to adopt existing schemes directly. The main technicality is that in the presence of a majority coalition, we cannot guarantee the liveness of the anonymous communication protocol.

To overcome this problem, one naïve idea is to rely on an anonymous communication protocol with identifiable abort, and if the protocol fails, we kick out an offending player and retry. Unfortunately, the vanilla notion of identifiable abort does not work for us because we cannot afford to kick out offending players one by one since we are aiming for small round complexity. Our idea is to devise an anonymous communication protocol not just with identifiable abort, but with with *plentiful identifiable aborts*. In other words, if the protocol fails, we want to kick out sufficiently many players, such that we can eventually succeed without too many retries.

Therefore, we adapt an anonymous communication protocol inspired by DC-nets [Cha88] to achieve such a plentiful identifiable abort notion. Assuming an upper bound of βn on the coalition size, our protocol kicks out at least $(1-\beta)n$ players in the event of failure. Thus the round complexity is at most $\frac{1}{1-\beta}$. For example, if $\beta = 99\%$, we can still succeed in $O(1)$ rounds.

We give a formal description of our poly-logarithmically-sized committee election protocol and prove its security in Section 4. We present a formal description of our anonymous communication protocol in Section 6.2.

2.3 Leader Election

Although the lightest bin protocol via anonymous broadcast (denoted as $\mathsf{LBin\text{-}V}$ below) achieves CSP-fairness and maximin-fairness simultaneously, it cannot be directly used to select a leader, i.e., $c = 1$. Indeed, the fairness of $\mathsf{LBin\text{-}V}$ depends on the occurrence of the good event that each bin has at least $(1-\epsilon^2)(1-\beta)c$ number of honest players, where $\beta \cdot n$ is the maximum coalition size for $\beta \in (0,1)$. If we are to choose a leader directly using $\mathsf{LBin\text{-}V}$, then the probability that this good event happens is 0, which makes our protocol unfair.

To construct a leader election protocol, we compose the committee election $\mathsf{LBin\text{-}V}$ for multiple iterations. In each iteration: we choose a log-sized committee. In the first iteration we choose a $\mathsf{poly}\log$-sized committee $\mathcal{C}_1$, and then in the second iteration we choose a $\mathsf{poly}\log\log$ sized committee $\mathcal{C}_2$ from $\mathcal{C}_1$, and so on. As analyzed earlier, each iteration of $\mathsf{LBin\text{-}V}$ is $(1-\Theta(\epsilon))$-game-theoretically fair given that the failure probability δ that the good event does not happen in this iteration is small compare to $\beta \cdot \epsilon$.

However, as the committee size becomes smaller in each iteration, the probability that the good event does not happen becomes larger. In the last few rounds, when the committee becomes constant size, the probability that the good event does not happen becomes a constant. Therefore, we need to cut off at some point and instead run the "almost perfect" tournament tree protocol. As shown in Chung et al. [CCWS21], the tournament tree protocol among c players chooses a leader in $O(\log c)$ rounds and is $(1-\mathsf{negl})$-game-theoretically fair. If we want to achieve a round complexity of R, then we can stop running $\mathsf{LBin\text{-}V}$ when the committee size becomes smaller than $2^{\Theta(R)}$ and run the tournament tree protocol among the committee to elect a leader.

Now suppose that we run L iterations of committee election $\mathsf{LBin\text{-}V}$ and get a committee of size $2^{\Theta(R)}$. Then we need to guarantee that the round complexity of these L iterations of $\mathsf{LBin\text{-}V}$ is at most $O(R)$. By the analysis above, if we kick out $(1-\beta)n$ players in each anonymous communication protocol, the round complexity of each $\mathsf{LBin\text{-}V}$ is at most $\frac{1}{1-\beta}$. This requires that the fraction of coalition $\beta \leq 1 - \frac{L}{\Theta(R)}$.

Now since the probability that the good event does not happen increases in each iteration, the probability that there is an iteration in which the good event does not happen is dominated by $L \cdot \delta_L$, where $\delta_L = \exp\{-\epsilon^4 \cdot 2^{-\Theta(R)}\}$ is the probability that good event does not happen in the last iteration. As long as this probability is smaller than $\beta \cdot \epsilon$, the protocol is $(1-\Theta(\epsilon))$-fair. Picking $\epsilon = \frac{1}{2^R}$ suffices. Therefore, if we run $\mathsf{LBin\text{-}V}$ multiple iterations to elect a committee $\mathcal{C}$ of size is $2^{\Theta(R)}$, and then run the tournament tree protocol among $\mathcal{C}$ to elect a leader, our leader election protocol achieves $(1-\frac{1}{2^{\Theta(R)}})$-game-theoretic fairness.

In Section 5, we give a generalized protocol that combines multiple iterations of $\mathsf{LBin\text{-}V}$ and the tournament tree protocol to elect an arbitrary-sized committee, including the special case of committee size 1, i.e., leader election.

3 Preliminaries

Notation. Throughout, we use λ to denote the security parameter. The notation $\log^{(\ell)} n$ means taking logarithm ℓ times over n. For example, $\log^{(3)} n \equiv \log\log\log n$. Moreover, we use $\log^* n$ to denote the smallest integer ℓ such that $\log^{(\ell)} n \leq 1$. For an event E, we denote $\overline{E}$ as the event that E does not happen. For a vector X of length M, we use $X[j]$ for $j \in [M]$ to denote the j-th element of X. By *t-out-of-n SS*, we refer to a Shamir secret sharing protocol in which any $t+1$ players can reconstruct the secret, while any t players know nothing about the secret [Sha79]. We use the acronym p.p.t. for non-uniform probabilistic polynomial time. We use $\{X_\lambda\}_\lambda \equiv_c \{Y_\lambda\}_\lambda$ to denote that two distribution ensembles $\{X_\lambda\}_\lambda$ and $\{Y_\lambda\}_\lambda$ are computationally indistinguishable, i.e., for all non-uniform p.p.t. $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$, such that for any $\lambda \in \mathbb{N}$, $|\Pr[x \overset{\$}{\leftarrow} X_n, \mathcal{A}(x) = 1] - \Pr[y \overset{\$}{\leftarrow} Y_n, \mathcal{A}(y) = 1]| < \mathsf{negl}(\lambda)$.

3.1 Probability Tools

Lemma 3.1 (Chernoff bound, Corollary A.1.14 [AS16])**.** *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables. Let $\mu = \mathbb{E}\left[\sum_{i=1}^n X_i\right]$. Then, for any $\epsilon \in (0,1)$, it holds that*

$$\Pr\left[\sum_{i=1}^n X_i \leq (1-\epsilon)\mu\right] \leq e^{-\epsilon^2\mu/2}.$$

3.2 Fairness Notions for Committee Election

Since a leader is a special case of a 1-sized committee, we will define correctness and fairness with respect to committee election protocol.

In a (c,n)-committee election protocol, n players interact through pairwise private channels and a public broadcast channel. We assume that each player has identity $1, 2, \ldots, n$, respectively. We assume that all communication channels are authenticated, i.e., messages carry the sender's identity. Moreover, the network is synchronous, and the protocol proceeds in rounds.

The protocol execution is parametrized with the security parameter λ. We assume that the coalition (adversary) A performs a *rushing* attack. In every round r, it waits for all honest players (those not in A) to send messages in round r and decide what messages the players in the coalition send in round r. At the end of the committee election, the protocol outputs a set of at most c players called the *committee*. The output is defined as a deterministic, polynomial-time function over all *public messages posted to the broadcast channel*. Since we assume that all players wish to be selected into the committee, the utility function we consider is as follows: each player elected into the committee gains a utility of 1, while everyone else gains a utility of 0. If all players behave honestly, the committee is chosen uniformly at random.

Correctness. We say that a (c, n)-committee election protocol is correct, if in an all honest execution, every subset $C \subset [n]$ of size c has an equal probability of being elected as the committee, where the probability is taken over the randomness of (an honest execution) the protocol.

For the fairness notion, we recall the definitions proposed by Chung et al. [CCWS21]. The first notion of fairness (CSP-fairness) protects against a malicious coalition from increasing its utility. The second notion (maximin-fairness) protects against a malicious coalition from decreasing the utility of any honest party. Each of these notions is natural and useful on its own, and in some sense, they complement each other. A protocol that satisfies both simultaneously is called *game-theoretically fair.*

Approximate CSP-fairness. The CSP-fairness requires that no coalition can increase its own expected utility by more than a $(1 - \epsilon)$ multiplicative factor, no matter how it deviates from the honest protocol.

Definition 3.2 ($(1 - \epsilon)$-CSP-fair committee election)**.** *A (c, n)-committee election is $(1 - \epsilon)$-CSP-fair against a non-uniform probabilistic polynomial time (p.p.t.) coalition A of size βn, iff no matter what strategy A adopts,*

$$\mathbb{E}[\widetilde{\beta}] \leq \frac{\beta}{1 - \epsilon},$$

where $\widetilde{\beta}$ is the fraction of players in the coalition among the committee, where the expectation is taken over the randomness of the protocol.

In our proof, we will also make use of another fairness notion:

Definition 3.3 ($(1 - \epsilon, \delta)$-CSP-fair committee election)**.** *A (c, n)-committee election is $(1 - \epsilon, \delta)$-CSP-fair against a non-uniform probabilistic polynomial time (p.p.t.) coalition A of size βn, if there exists an event GOOD, where $\Pr[\mathsf{GOOD}] \geq 1 - \delta$, such that no matter what strategy A adopts,*

$$\mathbb{E}[\widetilde{\beta} \mid \mathsf{GOOD}] \leq \frac{\beta}{1 - \epsilon},$$

where $\widetilde{\beta}$ is the fraction of the coalition's representation in the committee, and the expectation is taken over the randomness of the protocol.

Analogously, we define $(1-\epsilon)$-maximin-fair and $(1-\epsilon, \delta)$-maximin-fair committee election, which requires that the probability that an honest individual gets into the committee is large enough given that the good event happens.

Approximate maximin-fairness. Maximin-fairness requires that no coalition can harm any honest individual by more than a $(1-\epsilon)$ multiplicative factor, no matter how it deviates from the honest protocol.

Definition 3.4 ($(1-\epsilon)$-maximin-fair committee election)**.** *A (c,n)-committee election is $(1-\epsilon)$-maximin-fair against a non-uniform probabilistic polynomial time (p.p.t.) coalition A of size βn, iff for any honest individual i, the probability that i gets into the committee is*

$$\Pr[i \textit{ is in the committee}] \geq \frac{(1-\epsilon)c}{n},$$

no matter what strategy A adopts. The probability is taken over the randomness of the protocol.

Definition 3.5 ($(1-\epsilon,\delta)$-maximin-fairness)**.** *A (c,n)-committee election is $(1-\epsilon,\delta)$-maximin-fair against a non-uniform probabilistic polynomial time (p.p.t.) coalition A of size βn, if there exists an event GOOD, where $\Pr[\mathsf{GOOD}] \geq 1-\delta$, such that no matter what strategy A adopts,*

$$\Pr[i \textit{ is in the committee} \mid \mathsf{GOOD}] \geq \frac{(1-\epsilon)c}{n},$$

for any honest individual i. The probability is taken over the randomness of the protocol.

Although committee election is a constant-sum game, these two notions of fairness are non-equivalent. As shown in [CCWS21], approximate CSP-fairness and approximate maximin-fairness are different, although committee election is a constant-sum game. For example, in a $(1-o(1))$-CSP-fair (c,n) committee election protocol against a coalition A of size $0.9n$, the coalition may exclude a specific individual from being elected. Because the $\frac{c}{n}$ utility transferred from this honest individual to the coalition is very small compared to the coalition's default utility when playing honestly. On the other hand, in a $(1-O(1))$-maximin-fair (c,n) committee election against a small coalition A of size $O(1)$, the coalition can transfer $\frac{O(c)}{n}$ utility from each honest individual, and significantly increase its utility by a $\Theta(n)$ factor.

Finally, we define *game-theoretical fairness.* This notion of fairness requires CSP and maximin-fairness simultaneously.

Definition 3.6 ($(1-\epsilon)$-game-theoretical fairness)**.** *A (c,n)-committee election is $(1-\epsilon)$ game-theoretically fair committee election against a non-uniform probabilistic polynomial time (p.p.t.) coalition A, iff it is $(1-\epsilon)$-CSP-fair and $(1-\epsilon)$-maximin-fair against A.*

Definition 3.7 ($(1-\epsilon,\delta)$-game-theoretical fairness)**.** *A (c,n)-committee election is $(1-\epsilon)$ game-theoretically fair committee election against a non-uniform probabilistic polynomial time (p.p.t.) coalition A, iff it is $(1-\epsilon,\delta)$-CSP-fair and $(1-\epsilon,\delta)$-maximin-fair against A.*

By definition, a $(1-\epsilon)$-game-theoretically fair committee election is also $(1-\epsilon,0)$-game-theoretically fair. Next we give the translation from $(1-\epsilon,\delta)$-CSP/maximin-fair to $(1-\epsilon)$-CSP/maixin-fair.

Lemma 3.8. *Let n be the number of parties and fix a parameter c. Let CElect be an R-round $(1-\epsilon,\delta)$-CSP-fair (c,n)-committee election protocol against a coalition of size βn. Then the above leader election protocol is $(1-\epsilon_1)$-CSP-fair against a coalition of size βn, with a round complexity $R+O(\log c)$, where*

$$\epsilon_1 = \frac{\beta\epsilon + \delta(1-\epsilon)}{\beta + \delta(1-\epsilon)} + \mathsf{negl}(\lambda).$$

Proof. Let $\widetilde{\beta}$ denote the fraction of the coalition in committee $\mathcal{C}$. By Definition 3.5, there exists an event GOOD with $\Pr[\mathsf{GOOD}] \geq 1-\delta$, such that $\mathbb{E}[\widetilde{\beta} \mid \mathsf{GOOD}] \leq \frac{\beta}{1-\epsilon}$. By the law of total expectation,

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\beta}\right] &= \mathbb{E}\left[\widetilde{\beta} \mid \mathsf{GOOD}\right] \cdot \Pr\left[\mathsf{GOOD}\right] + \mathbb{E}\left[\widetilde{\beta} \mid \overline{\mathsf{GOOD}}\right] \cdot \Pr\left[\overline{\mathsf{GOOD}}\right] \\
&\leq \frac{\beta}{1-\epsilon} + (1 - \Pr[\mathsf{GOOD}]) \\
&\leq \frac{\beta}{1-\epsilon} + \delta \\
&= \beta \cdot \frac{\beta + \delta(1-\epsilon)}{\beta(1-\epsilon)}.
\end{aligned}$$

Combine with Lemma 5.1, the expected utility of the coalition is at most $\beta \cdot \frac{\beta+\delta(1-\epsilon)}{\beta(1-\epsilon)(1-\mathsf{negl}(\lambda))} \leq \frac{\beta}{1-\epsilon_1}$, and the round complexity is $R + O(\log c)$. The lemma thus follows. □

Lemma 3.9. *Let n be the number of parties and fix a parameter c. Let CElect be an R-round $(1-\epsilon, \delta)$-maximin-fair (c, n)-committee election protocol against a coalition of size βn. Then the above leader election protocol is $(1-\epsilon_2)$-maximin-fair, with a round complexity $R + O(\log c)$, where*

$$\epsilon_2 = \epsilon + \delta + \mathsf{negl}(\lambda).$$

Proof. Let H_i denote the event that honest player i gets elected into the committee $\mathcal{C}$. By Definition 3.7, there exists an event GOOD with $\Pr[\mathsf{GOOD}] \geq 1-\delta$, such that $\Pr[\mathsf{H}_i \mid \mathsf{GOOD}] \geq \frac{(1-\epsilon)c}{n}$. By the law of total probability,

$$\begin{aligned}
\Pr\left[\mathsf{H}_i\right] &\geq \Pr\left[\mathsf{H}_i \mid \mathsf{GOOD}\right] \Pr\left[\mathsf{GOOD}\right] \\
&\geq \frac{(1-\epsilon)c}{n} \cdot (1-\delta) \geq \frac{(1-\epsilon-\delta)c}{n}.
\end{aligned}$$

Combine with Lemma 5.1, the probability that an honest player gets elected as the leader is at least $\frac{(1-\epsilon-\delta)c}{n} \cdot \frac{1-\mathsf{negl}(\lambda)}{2} \geq \frac{1-\epsilon_2}{n}$, and the round complexity is $R + O(\log c)$. The lemma thus follows. □

Hybrid vs. real worlds. For ease of presentation and modulatiry purposes, we shall sometimes consider protocols in a hybrid setting where we assume some "generic" functionality is given for free. This is called a "hybrid world". That is, we say that a protocol is in the $\mathcal{F}$-hybrid world if players interacting in this protocol have access to an ideal functionality $\mathcal{F}$. A protocol in the (plain) real world is a protocol without any ideal functionalities or setup assumptions. Specifically for us, we say that a (c, n)-committee election protocol achieves $(1-\epsilon)$-game-theoretic fairness against a coalition A in the $\mathcal{F}$-hybrid world, if the protocol achieves $(1-\epsilon)$-game-theoretic fairness against this coalition A, assuming the ideal functionality $\mathcal{F}$.

3.3 Publicly Verifiable Concurrent Non-Malleable Commitment

A publicly verifiable commitment scheme $(\mathsf{C}, \mathsf{R}, \mathsf{V})$ consists of a pair of interacting Turing machines, the committer C, the receiver R, and a deterministic, polynomial-time public verifier V. We assume that the protocol has two phases, a commitment phase and an opening phase. The public verifier, upon receiving a transcript Γ of the commitment protocol, outputs either a bit $b \in \{0, 1\}$ to accept or $\perp$ to reject. We use $\langle \mathsf{C}^*(z), \mathsf{R}^*(z') \rangle$ to denote an execution between C^* on input $z, 1^\lambda$, and R^* on input $z', 1^\lambda$, where λ is the security parameter.

Correctness. Correctness guarantees that an honest committer always completes the protocol and correctly opens its input bit; and will not be stuck by a malicious, non-aborting receiver. Formally, for $b \in \{0,1\}$, for any $\lambda \in \mathbb{N}$, if C is honest and receives input bit b, then $\langle \mathsf{C}(z), \mathsf{R}^*(z') \rangle$ will complete with the accepting bit b with probability 1, for any non-aborting R^*. If the messages sent by R^* are outside the valid range, it is treated as aborting.

Perfect Binding. Perfect binding guarantees that the commitment phase will determine only one bit that can be successfully opened. Formally, let $(\Gamma_c, \Gamma_o) \in \{0,1\}^{\ell(\lambda)}$ be the transcripts of the commitment phase and the opening phase, respectively, where $\ell(\lambda)$ is a fixed polynomial function denoting the maximum length of the transcripts. Then for any $\lambda \in \mathbb{N}$, any transcripts $\Gamma_c, \Gamma_o, \Gamma'_o$, if $\mathsf{V}(1^\lambda, \Gamma_c, \Gamma_o) = b$ and $\mathsf{V}(1^\lambda, \Gamma_c, \Gamma'_o) = b'$, where $b, b' \in \{0,1\}$, it must be that $b = b'$.

Computationally Hiding. Computationally hiding guarantees that at the end of the commitment phase, the receiver learns only a negligible amount of information about the input that the committer commits to. Formally, let $p_\lambda(v)$ denote the probability that R^* outputs 1 at the end of the commitment phase in an execution $\langle \mathsf{C}^*(1^\lambda, v), \mathsf{R}^*(1^\lambda) \rangle$, then for any non-uniform p.p.t. R^*, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$ and every $v_1, v_2 \in \{0,1\}^\lambda$, it holds that $|p_\lambda(v_1) - p_\lambda(v_2)| \leq \mathsf{negl}(\lambda)$.

Concurrent Non-malleability. We follow the definition of Lin et al. [LPV08]. Consider a man-in-the-middle adversary A that participate on the left m interactions with an honest committer who runs commitment phase committing to values $v_1, \ldots, v_m$ with identity $\mathsf{id}_1, \ldots, \mathsf{id}_m$, and on the right m interactions with an honest receiver trying to commit to values $v'_1, \ldots, v'_m$ with identity $\mathsf{id}'_1, \ldots, \mathsf{id}'_m$. If any of the right commitments are invalid its value is set to $\perp$. For every $i \in [m]$, if $\mathsf{id}'_j = \mathsf{id}_i$ for some $j \in [m]$, then v'_j is set to be $\perp$. Let $\mathsf{mitm}^{\mathcal{A}}(1^\lambda, v_1, v_2, \ldots, v_m, z)$ denote the view of $\mathcal{A}$ and the values $v'_1, \ldots, v'_m$.

Definition 3.10. *A commitment scheme is concurrent non-malleable if for every polynomial $p(\cdot)$, for every non-uniform p.p.t. adversary $\mathcal{A}$ that participates in at most $m = p(\lambda)$ concurrent executions, there exists a polynomial time simulator $\mathcal{S}$ such that*

$$\{\mathsf{mitm}^{\mathcal{A}}(1^\lambda, v_1, v_2, \ldots, v_m, z)\}_{v_1,\ldots,v_m \in \{0,1\}, z \in \{0,1\}^*, \lambda \in \mathbb{N}} \equiv_c$$
$$\{\mathcal{S}(1^\lambda, z)\}_{v_1,\ldots,v_m \in \{0,1\}, z \in \{0,1\}^*, \lambda \in \mathbb{N}}.$$

Theorem 3.11 ([LPV08])**.** *Assume that one-way permutations exist. Then there exists a constant-round, publicly verifiable commitment scheme that is perfectly correct, perfectly binding, and concurrent non-malleable.*

In this paper, we will only consider bounded concurrency. Without loss of generality, the number of concurrent calls to public verifiable concurrent non-malleable commitment in our protocol is upper bounded by n^2, where n is the number of players.

4 Game-Theoretically Fair Committee Election

In this section, we present our game-theoretically fair committee election that extends Feige's lightest bin protocol. Later, in Section 5, we will use it as a building block to get our committee election protocol that achieves game-theoretic fairness for arbitrary committee size.

4.1 Electing Poly-logarithmically Sized Committees: Achieving CSP-Fairness

In this section, we give a CSP-fair committee election protocol. This is the first step towards our game-theoretically fair committee election (that needs to be CSP-fair and maximin fair, simultaneously).

Our CSP-fair protocol is a commit-and-reveal variant of Feige's well-known lightest bin protocol [Fei99]. Specifically, we require all parties to (cryptographically) commit to their bin choices and only afterward to reveal their choices. The parties whose choices correspond to the lightest bin are the committee. The commitments that we use are *interactive*. To commit to a string, a player invokes n instances of NMC, one for each of the n receivers. To open the commitments, the committer posts the openings for all n instances in the broadcast channel, and the opening is correct iff all of the n instances are correctly opened to the same string. Without loss of generality, we assume that the committer only needs to send one message in the opening phase. Moreover, we assume that messages are posted to the broadcast channel, and it can be checked publicly if a commitment is correctly opened. This is why we also require public verifiability of the commitment scheme. We say that a player fails to commit if the player fails to commit in an instance, where the receiver is non-aborting.

$\mathsf{LBin\text{-}C}$: **Commit-and-Reveal Lightest Bin**

Parameters: Let c be an upper bound of the size of the required committee and n is the number of players. Fix $B = \lceil \frac{n}{c} \rceil$ as the number of bins. For simplicity, we assume c divides n.

Building blocks: A publicly verifiable concurrent non-malleable commitment as in Section 3.3, NMC.

Protocol:

1. Round 1: Every player i randomly chooses a bin $b_i \in [B]$, invokes n NMC instances and run the commit phase with n receivers to commit to b_i. The messages are sent in a broadcast channel. Exclude those players who fail to commit.
2. Round 2: Every player i runs the opening phase with n receivers to open its bin choice b_i. Exclude those players who fail to open all n instances correctly.
3. Let $\widehat{b}$ be the lightest bin after exclusion (break ties with lexicographically the smallest bin). The players who choose bin $\widehat{b}$ constitute the committee.

Theorem 4.1. *Assume that* NMC *is publicly verifiable concurrent non-malleable commitment as in Section 3.3. For* $n, c \in \mathbb{N}$, $\epsilon \in (0, 1/2)$, *and* $\beta \in (0, 1)$, *the protocol* $\mathsf{LBin\text{-}C}$ *is a constant round* $(1 - 2\epsilon, \delta)$*-CSP-fair* (c, n)*-committee election protocol against a coalition* $\mathcal{K}$ *of size* βn*, where*

$$\delta = \frac{n}{c} \exp\left\{ -\frac{\epsilon^4}{2}(1-\beta)c \right\}. \tag{1}$$

Proof. Fix n, c, ϵ, and β as in the statement. Define GOOD to be the event that each bin has at least $(1-\epsilon^2)(1-\beta)c$ honest players. Let $\widetilde{\beta}$ denote the fraction of players in $\mathcal{K}$ among the committee. Then, we have the following lemma.

Lemma 4.2. $\mathbb{E}\left[\widetilde{\beta} \mid \mathsf{GOOD}\right] \leq \frac{\beta}{1-2\epsilon}$.

For now assume that Lemma 4.2 holds and we explain why Theorem 4.1 follows from it. The proof of Lemma 4.2 appears right afterwards. Let $X_{i,b}$ be an indicator random variable that honest player i chooses bin $b \in [B]$. Then, $X_{i,b}$ is a Bernoulli random variable that takes 1 with probability c/n. Since the number of honest players is $n - \beta n$, by linearity of expectation, $\mathbb{E}[\sum_{i \in \mathcal{H}} X_{i,b}] = (1-\beta)c$, where $\mathcal{H}$ denotes the set of honest players. For a fixed bin b, by Chernoff bound (Lemma 3.1), the probability that the number of honest players choosing this bin is less than $(1-\epsilon^2)(1-\beta)c$ is

$$\Pr\left[\sum_{i \in \mathcal{H}} X_{i,b} \le (1-\epsilon^2)(1-\beta)c\right] \le \exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\}.$$

By the union bound over the B bins, the probability that GOOD happens is at least

$$\Pr[\mathsf{GOOD}] \ge 1 - \frac{n}{c}\exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\}. \tag{2}$$

Combing Lemma 4.2 and (2), $\mathsf{LBin\text{-}C}$ is a $(1-2\epsilon, \delta)$-CSP-fair committee election protocol by Definition 3.5. □

We now proceed with the proof of Lemma 4.2.

Proof of Lemma 4.2. We split into two cases. First, assume that $\beta \ge \epsilon$. In this case, the claim follows directly from the assumption that GOOD holds. Specifically, since each bin contains at least $(1-\epsilon^2)(1-\beta)c$ honest players, the committee contains at least $(1-\epsilon^2)(1-\beta)$ fraction of honest players. It follows that the fraction of players in $\mathcal{K}$ among the committee must satisfy

$$\widetilde{\beta} \le 1 - (1-\epsilon^2)(1-\beta) = \beta\left(1 + \frac{\epsilon^2}{\beta} - \epsilon^2\right) \le \frac{\beta}{1-2\epsilon},$$

as required.

Now, we focus on the case where $\beta < \epsilon$. In this case, the proof relies on the concurrent non-malleability of the commitment scheme. Specifically, we consider the following hybrid experiment.

Hybrid experiment Hyb: the hybrid experiment essentially runs $\mathsf{LBin\text{-}C}$ but the bin choices of the coalition are chosen by the NMC's simulator $\mathcal{S}$. Recall that the simulator $\mathcal{S}$ outputs at most n^2 values that the coalition commits to, as well as the view of the coalition in the man-in-the-middle game.

Hyb: Hybrid experiment

1. Each honest player randomly chooses a bin $b_i \in [B]$, invokes n instances of NMC and run the commit phase with n receivers to commit to b_i.
2. Run the simulator $\mathcal{S}$ for NMC, that outputs 1) βn^2 values $\{b_j^1, \dots, b_j^n\}_{j \in \mathcal{K}}$ that the coalition is trying to commit: each player $j \in \mathcal{K}$ commits to b_j^i to receiver i, for $i \in [n]$; and 2) the view of the adversary view. If the same player j is committing to different values to different receivers, we simply let its committed value be $b_j = 0$; otherwise, we let b_j, i.e., the bin choice of player j, be the value output by the simulator $\mathcal{S}$.

Output: The experiment outputs the bin choices of each players $b_1, \dots, b_n$.

Note that the outputs $b_1, \dots, b_n$ are not efficiently computable but are well-defined due to the perfect binding property of NMC. The hybrid experiment Hyb stops before the opening phase of the NMC, because by the concurrent non-malleability definition (Section 3.3), the simulatability of the NMC only holds for the commitment phase.

We use Real to denote a real execution of Round 1 of $\mathsf{LBin\text{-}C}$, i.e., every player commits to its bin choice by invoking n instances of NMC and running the commit phase with n receivers. The output is the bin choice of each player. Still, the outputs $b_1, \ldots, b_n$ are not efficiently computable but are well-defined due to the perfect binding property of NMC.

To compute $\mathbb{E}[\widetilde{\beta} \mid \mathsf{GOOD}]$, we define a random variable γ, which depends only on $\{b_i\}_{i=1}^n$, that upper bounds $\widetilde{\beta}$ in an execution of $\mathsf{LBin\text{-}C}$. Let $\widetilde{b} \in [B]$ be the index of the bin that contains least number of honest players; and $b^* \in [B]$ be index of the lightest bin at the end of the commit phase. Note that by the way the protocol works, $\widetilde{b}$ and b^* depends only on $\{b_i\}_{i=1}^n$. Below, for $l \in [B]$, we use h_l to denote the number of honest players in bin l, and f_l to denote the number of players in $\mathcal{K}$ in bin l.

Claim 4.3. *Given the bin choices $\{b_i\}_{i=1}^n$ at the end of the commit phase, the fraction of players in $\mathcal{K}$ among the committee $\widetilde{\beta}$ is at most $\gamma := \frac{f_{\widetilde{b}}}{h_{b^*}+f_{b^*}}$.*

Proof. Given the bin choices at the end of the commit phase $\{b_i\}_{i=1}^n$, by the perfect binding property of NMC, a player $j \in \mathcal{K}$ can either choose to open the correct bin choice b_j it commits to, or fail to open its bin choice in the opening phase. If some players in $\mathcal{K}$ refuse to open their commitments, by the public verifiability of NMC, they will be excluded at the end of the opening phase, which may change the lightest bin. That is to say, after the commit phase, the only way the coalition can deviate is essentially to refuse to open some of their bin choices, in order to change the lightest bin.

If the some players in $\mathcal{K}$ refuse to open their commitments and make bin k the lightest bin, then after excluding those misbehaved players, the number of players in bin k is at most $f_{b^*} + h_{b^*}$ (otherwise bin b^* is still the lightest bin). Since the number of honest players in bin l is h_l, the fraction of players in $\mathcal{K}$ in bin l, after excluding the misbehaved players, is at most $1 - \frac{h_l}{f_{b^*}+h_{b^*}}$. To maximize the fraction of the coalition in the committee, the best strategy for the coalition is to choose bin $l = \widetilde{b}$, which contains the least number of honest players.

Therefore, the best strategy for the malicious coalition is to make bin $\widetilde{b}$ the lightest bin at the end of the opening phase. The maximum fraction of the coalition in the committee is thus bounded by

$$\widetilde{\beta} \leq 1 - \frac{h_{\widetilde{b}}}{h_{b^*} + f_{b^*}} \leq \frac{f_{\widetilde{b}}}{h_{b^*} + f_{b^*}},$$

where the inequality follows from the fact that the number of the coalition's representation in bin $\widetilde{b}$ is at most $f_{\widetilde{b}}$. □

Therefore, to upper bound $\mathbb{E}[\widetilde{\beta} \mid \mathsf{GOOD}]$, it suffices to bound $\mathbb{E}[\gamma \mid \mathsf{GOOD}]$ in the Real experiment, which only depends on $\{b_i\}_{i=1}^n$.

We then argue the computational indistinguishability of γ in Hyb and in Real. If this holds, then we only need to bound $\mathbb{E}[\gamma \mid \mathsf{GOOD}]$ in the Hyb experiment.

Claim 4.4. *The distribution of γ in the real experiment* Real *(denoted as γ_{real}) is computationally indistinguishable from that of γ in the hybrid experiment* Hyb *(denoted as γ_{Hyb}).*

Proof. Suppose for the sake of contradiction that there exists a non-uniform p.p.t. adversary $\mathcal{D}$ that can distinguish γ_{Real} and γ_{Hyb}.

Consider the following hybrid experiment Hyb', which is same as Hyb except that the bin choices of players in the coalition are chosen by $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, 0, \ldots, 0, z)$. Since in both Hyb and Hyb', the coalition's bin choices are chosen independently from honest players' bin choices, by the non-malleability of NMC, $\mathsf{Hyb} \equiv_c \mathsf{Hyb}'$. Therefore, the random variable γ_{Hyb} should be indistinguishable from γ in Hyb', denoted as $\gamma_{\mathsf{Hyb}'}$. This means that $\mathcal{D}$ should be able to distinguish γ_{Real} from $\gamma_{\mathsf{Hyb}'}$ with a non-negligible probability.

Now we can construct a non-uniform p.p.t. adversary $\mathcal{D}'$ that can distinguish $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, \{b_i\}_{i\in\mathcal{H}}, z)$ from $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, 0, \ldots, 0, z)$. $\mathcal{D}'$ works as follows: it randomly picks $\{b_i\}_{i\in\mathcal{H}}$ as the bin choices each honest player commits to, and send $\{b_i\}_{i\in\mathcal{H}}$ to the challenger. Upon receiving the bin choices for players in the coalition from the challenger, which is either output from $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, \{b_i\}_{i\in\mathcal{H}}, z)$, or $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, 0, \ldots, 0, z)$, it computes γ given the bin choices of every player. $\mathcal{D}'$ then passes γ to $\mathcal{D}$ and output whatever $\mathcal{D}$ outputs.

If the coalition's bin choices are generated from $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, \{b_i\}_{i\in\mathcal{H}}, z)$, then $\mathcal{D}$'s view is same as γ_{Real}; otherwise, $\mathcal{D}$'s view is same as $\gamma_{\mathsf{Hyb}'}$. This implies that $\mathcal{D}'$ can distinguish $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, \{b_i\}_{i\in\mathcal{H}}, z)$ from $\mathsf{mimt}^{\mathcal{A}}(1^\lambda, 0, \ldots, 0, z)$ with a non-negligible probability. This contradicts the non-malleability of NMC that there exists a simulator $\mathcal{S}$, such that

$$\mathsf{mimt}^{\mathcal{A}}(1^\lambda, \{b_i\}_{i\in\mathcal{H}}, z) \equiv_c \mathcal{S}(1^\lambda, z) \text{ and}$$
$$\mathsf{mimt}^{\mathcal{A}}(1^\lambda, 0, \ldots, 0, z) \equiv_c \mathcal{S}(1^\lambda, z)$$

To conclude, γ_{Real} is computationally indistinguishable from γ_{Hyb}. □

Now, it suffices to bound $\mathbb{E}[\gamma \mid \mathsf{GOOD}]$ in the hybrid experiment Hyb.

Claim 4.5. *In the hybrid experiment,* $\mathbb{E}\left[\gamma \mid \mathsf{GOOD}\right] \le \frac{\beta}{(1-\epsilon^2)(1-\epsilon)}$.

Proof. In Hyb, the coalition $\mathcal{K}$'s bin choices are chosen by the simulator $\mathcal{S}$, who has no access to the bin choice of honest players. Therefore, the malicious coalition's bin choices $\{b_j\}_{j\in\mathcal{K}}$ are independent of honest players' bin choices $\{b_i\}_{i\in\mathcal{H}}$, where $\mathcal{K}$ and $\mathcal{H}$ denote the coalition and the set of honest players, respectively. This implies that the number of the coalition's representation f_l in bin l is independent from $\widetilde{b}$, which depends only on $\{b_i\}_{i\in\mathcal{H}}$. Moreover, f_l is independent from GOOD since GOOD also depends only on honest players bin choices.

As a consequence, by the law of total expectation,

$$\begin{aligned}
&\mathbb{E}[\gamma \mid \mathsf{GOOD}] \\
&= \sum_{l=1}^{B}\sum_{l'=1}^{B} \mathbb{E}\left[\gamma \mid \widetilde{b} = l, b^* = l', \mathsf{GOOD}\right] \Pr\left[\widetilde{b} = l, b^* = l' \mid \mathsf{GOOD}\right] \\
&= \sum_{l=1}^{B}\sum_{l'=1}^{B} \mathbb{E}\left[\gamma \mid \widetilde{b} = l, b^* = l', \mathsf{GOOD}\right] \Pr\left[b^* = l' \mid \widetilde{b} = l, \mathsf{GOOD}\right] \Pr\left[\widetilde{b} = l \mid \mathsf{GOOD}\right] \\
&= \frac{c}{n}\sum_{l=1}^{B}\sum_{l'=1}^{B} \mathbb{E}\left[\frac{f_l}{h_{l'} + f_{l'}} \mid \widetilde{b} = l, b^* = l', \mathsf{GOOD}\right] \Pr\left[b^* = l' \mid \widetilde{b} = l, \mathsf{GOOD}\right]. \qquad (3)
\end{aligned}$$

Since when GOOD happens, the number of honest players in every bin is at least $(1-\epsilon^2)(1-\beta)c$, we have that $\frac{f_l}{h_{l'}+f_{l'}} \le \frac{f_l}{(1-\epsilon^2)(1-\beta)c}$ for any $l, l' \in [B]$. Thus, (3) is at most

$$\begin{aligned}
(3) \le &\frac{1}{(1-\epsilon^2)(1-\beta)n}\sum_{l=1}^{B}\left(\sum_{l'=1}^{B} \mathbb{E}\left[f_l \mid \widetilde{b} = l, b^* = l', \mathsf{GOOD}\right] \Pr\left[b^* = l' \mid \widetilde{b} = l, \mathsf{GOOD}\right]\right) \\
= &\frac{1}{(1-\epsilon^2)(1-\beta)n}\sum_{l=1}^{B} \mathbb{E}\left[f_l \mid \widetilde{b} = l, \mathsf{GOOD}\right].
\end{aligned}$$

Since f_l is independent of $\widetilde{b}$ and GOOD, we have that

$$\mathbb{E}[\gamma \mid \mathsf{GOOD}] = \frac{1}{(1-\epsilon^2)(1-\beta)n}\sum_{l=1}^{B}\mathbb{E}[f_l] = \frac{\beta}{(1-\epsilon^2)(1-\beta)} \le \frac{\beta}{(1-\epsilon^2)(1-\epsilon)},$$

where the last inequality comes from the assumption that $\beta < \epsilon$. □

Putting together, the expectation $\mathbb{E}\left[\widetilde{\beta} \mid \mathsf{GOOD}\right]$ in the committee election $\mathsf{LBin\text{-}C}$ is at most $\frac{\beta}{(1-\epsilon^2)(1-\epsilon)} + \mathsf{negl}(\lambda) \le \frac{\beta}{1-2\epsilon}$. □

Corollary 4.6. $\Pr[\widetilde{\beta} \le \beta(1-\epsilon^2)+\epsilon^2 \mid \mathsf{GOOD}] = 1$.

Proof. This is because when GOOD happens, the fraction of honest players in the committee is at least $(1-\epsilon)(1-\beta)$. The fraction of the coalition is at most $1-(1-\epsilon)(1-\beta) = \beta(1-\epsilon^2)+\epsilon^2$, i.e., $\Pr[\widetilde{\beta} \le \beta(1-\epsilon^2)+\epsilon^2 \mid \mathsf{GOOD}] = 1$. □

4.2 Electing Poly-logarithmically Sized Committees: Achieving Maximin-Fairness

In Section 4.1 we gave a commit-and-reveal variant of Feige's lightest bin protocol for committee election and showed that it is CSP-fair. The protocol is, however, not maximin-fair. While the adversary cannot gain too much utility by deviating from the protocol, it can still harm the utility of an honest individual. Specifically, consider the following adversarial strategy. The coalition generates commitments so that the coalition's representations in each bin are equal. Then, when it wants to target at a specific player i to not participate in the committee, it waits to see which bin l was chosen by that honest party and then it refuses to reveal commitments from some other bin l' which will then be lighter than the bin l chosen by honest player i. This attack prevents an honest individual i from being elected into the committee.

By the properties of the commitment scheme and how our protocol works, this is the only useful attack for the adversary. Thus, we modify our protocol to withstand this attack by masking the identity of parties. Namely, we hide which bin choice belongs to which party. We achieve this by requiring players to choose a random virtual ID and use it throughout the execution. Players will only reveal their virtual IDs at the end of the protocol, after the lightest bin has been fixed. A-priori, it seems hard to implement such a system because once a party sends its message, everybody knows who sent it (recall that we are in the broadcast model). We overcome this by implementing an "anonymous" broadcast channel on top of our existing broadcast channel.

Thus, we first describe our anonymous broadcast functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Then, we show that in a $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$-hybrid model, we can build a committee election protocol that ensures CSP-fairness and maximin-fairness simultaneously.

4.2.1 Anonymous Broadcast Functionality

Let $\mathcal{O}$ be the set of all players involving in the protocol. Our anonymous broadcast functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ works as follows.

$\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$: Anonymous broadcast with t-identifiable abort

Parameters: $\mathcal{O}$ is the set of players involving in the protocol and t is a bound on the number of misbehaved players to exclude.

Functionality:

1. **Input**: Every player i sends a single message m_i or $\perp$ to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$.

2. **Output**: $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ computes a multiset $\mathsf{Out} = \{m_i : i \in \mathcal{O} \text{ and } m_i \neq \perp\}$.

   If the number of corrupted players is smaller than t, send $(\mathsf{ok}, \mathsf{Out})$ to everyone in $\mathcal{O}$. Otherwise, send Out to the adversary $\mathcal{A}$.

   - If receives ok from $\mathcal{A}$, $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ sends $(\mathsf{ok}, \mathsf{Out})$ to every honest player in $\mathcal{O}$.
   - Otherwise, it receives a set $\mathcal{D}$ of corrupted IDs of size at least t from the adversary $\mathcal{A}$, and then send $(\mathsf{fail}, \mathcal{D})$ to every honest player in $\mathcal{O}$.

We say that an adversary A is *admissible* if 1) it sends only one message for each corrupt player, and 2) it either sends ok, or a set of corrupted players of size at least t in Step 2.

The functionality exhibits several appealing properties that are important for us. Specifically, in the ideal functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, it holds that:

1. Each player can only send one message.
2. The coalition has to choose their messages independently from honest players' messages.
3. The coalition cannot tell which honest player sends which message.
4. The output is either $(\mathsf{ok}, \mathsf{Out})$, or $(\mathsf{fail}, \mathcal{D})$ with a set $\mathcal{D}$ of size at least t.

4.2.2 Formal Description of the Protocol

Here we present the formal description of our lightest bin via anonymous broadcast protocol in the $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$-hybrid model.

$\mathsf{LBin\text{-}V}(c, n, \beta)$: **Lightest Bin via Anonymous Broadcast**

Parameters: Let c be an upper bound of the required committee and n is the number of players. Fix $B = \lceil \frac{n}{c} \rceil$ as the number of bins. For simplicity, we assume c divides n. Let $\mathcal{O}$ be initialized as $[n]$ that denotes the set of active players. $\beta \cdot n$ is the maximum size of the coalition for $\beta \in (0, 1)$.

Building blocks: A publicly verifiable concurrent non-malleable commitment as in Section 3.3, NMC.

Protocol:

1. Every player i randomly chooses a string $v_i \leftarrow \{0,1\}^\lambda$ as its virtual ID, invokes n instances of NMC, and runs the commit phase with n receivers to commit to (i, v_i). Exclude those players who fail to commit.

2. Each player randomly chooses a bin $b_i \leftarrow [B]$ with fresh randomness, and sets $m_i = (b_i, v_i)$. Broadcast m_i using $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ with $t = \lfloor (1-\beta)n \rfloor$.

   - If the output is $(\mathsf{fail}, \mathcal{D})$, exclude the players in $\mathcal{D}$ from $\mathcal{O}$ (namely, set $\mathcal{O} = \mathcal{O} \setminus \mathcal{D}$). Then, the remaining players (i.e., those in the updated $\mathcal{O}$) re-run step 2.
   - If the output is $(\mathsf{ok}, \mathsf{Out})$, go to the next step.

3. Let b^* be the lightest bin. Every player opens its virtual ID (i, v_i). Let U_{b^*} be the set of virtual IDs that are *unique* and choose the lightest bin b^*. Those who open the (i, v_i) successfully with $v_i \in U_{b^*}$ are chosen to be the committee.

Note that in LBin-V, players do not need to commit to their bin choices and then open, since the functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ guarantees that the malicious coalition has to choose their messages, i.e., bin choices, independently from honest players' messages. In the following theorem we show that the protocol LBin-V described above is both maximin-fair and CSP-fair in the $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$-hybrid model.

Theorem 4.7. *Assume that* NMC *is a publicly verifiable concurrent non-malleable commitment as in Section 3.3. For any $n, c \in \mathbb{N}$ and $\epsilon \in (0, 1/2), \beta \in (0, 1)$, the committee election protocol* LBin-V(c, n, β) *is a $(1-\epsilon, \delta)$-maximin-fair and a $(1-2\epsilon, \delta)$-CSP-fair (c, n)-committee election* [2] *in the $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$-hybrid model, against a coalition $\mathcal{K}$ of size βn, where*

$$\delta = \frac{2n}{(1-\beta)c} \exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\} + \mathsf{negl}(\lambda).$$

Moreover, the round complexity of LBin-V *is at most $\frac{2}{1-\beta} + 2$.*

Proof. Fix n, c, ϵ, and β as in the statement. Let Unique be the event that honest players choose unique virtual IDs, and their virtual IDs do not collide with any players in the coalition. Let GOOD be the event that in every execution of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in Step 2, each bin has at least $(1-\epsilon^2)(1-\beta)c$ honest players.

We use the following lemma to prove maximin-fairness and CSP-fairness. The proof to the lemma appears afterward.

Lemma 4.8. $\Pr[\mathsf{Unique}, \mathsf{GOOD}] \geq 1 - \delta$.

Maximin-fairness Let H_i denote the event that an honest player i is chosen into the committee. The claimed maximin-fairness follows from the following lemma. The proof of the lemma appears below.

Lemma 4.9. $\Pr[\mathsf{H}_i \mid \mathsf{Unique}, \mathsf{GOOD}] \geq (1-\epsilon)c/n$.

Combining Lemmas 4.8 and 4.9, we have that LBin-V is a $(1-\epsilon, \delta)$-maximin-fair committee election protocol against a coalition of size βn by Definition 3.7.

CSP-fairness Let $\widetilde{\beta}$ denote the fraction of the coalition in the committee. Now, the claimed CSP-fairness follows from the following lemma. The proof of the lemma appears below.

Lemma 4.10. $\mathbb{E}\left[\widetilde{\beta} \mid \mathsf{GOOD}, \mathsf{Unique}\right] \leq \frac{\beta}{1-2\epsilon}$.

Combining Lemmas 4.8 and 4.10, we have that LBin-V is a $(1-2\epsilon, \delta)$-CSP-fair committee election protocol against a coalition of size βn by Definition 3.5. □

4.2.3 Proof of Lemma 4.8

Since Unique depends only on the virtual IDs $(v_1, \ldots, v_n)$, Consider the following hybrid experiment.

[2] Theorem 4.7 implies that the protocol LBin-V is a $(1-2\epsilon, \delta)$-game-theoretic fairness by Definition 3.7.

Hyb: Hybrid experiment

1. Each honest player randomly chooses a virtual ID $v_i \in \{0,1\}^\lambda$, invokes n instances of NMC and run the commit phase with n receivers to commit to v_i.
2. Run the simulator $\mathcal{S}$ for NMC, that outputs 1) βn^2 values $\{v_j^1, \ldots, v_j^n\}_{j\in\mathcal{K}}$ that the malicious coalition is trying to commit: each player $j \in \mathcal{K}$ commits to v_j^i to receiver i, for $i \in [n]$; and 2) the view of the adversary view_A. If the same player $j \in \mathcal{K}$ is committing to different values to different receivers, we simply let its committed value be $v_j = 0^\lambda$; otherwise, we let v_j, i.e., the virtual ID of player $j \in \mathcal{K}$, be the value output by the simulator $\mathcal{S}$.

Output: The experiment outputs the set of virtual IDs of honest players V_H and the set of virtual IDs of the coalition V_K, as well as the view of the malicious coalition view_K.

We use Real to denote a real execution of Step 1 of $\mathsf{LBin\text{-}V}$. The output is the set of virtual IDs chosen by honest players, players in the coalition, and the view of the adversary. By a similar argument as Claim 4.4, $\Pr[\mathsf{Unique}]$ in Real should be negligibly close to $\Pr[\mathsf{Unique}]$ in Hyb. Note that in Hyb, the probability that an honest player i chooses a virtual ID that collides with another player j with probability at most $\frac{1}{2^\lambda}$. By the union bound over the number of pairs of players, Unique happens with $1 - \frac{n(n-1)}{2\cdot 2^\lambda}$ probability in Hyb. Therefore, in Real, the probability $\Pr[\mathsf{Unique}] \geq 1 - \mathsf{negl}(\lambda)$.

By Chernoff's bound (Lemma 3.1) and the union bound over B bins, in a single execution of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, each bin contains at least $(1-\epsilon^2)(1-\beta)c$ honest players with probability

$$p = 1 - \frac{n}{c}\exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\}.$$

Next, we argue that the number of executions of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in $\mathsf{LBin\text{-}V}$ is at most $\frac{2}{1-\beta}$. Indeed, each time we invoke $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, the protocol either outputs ok or outputs a set of players in the coalition of size at least t. These t players in the coalition are excluded from the later executions of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Since $t = \lfloor (1-\beta)n \rfloor$, we will run at most $\frac{\beta n}{\lfloor (1-\beta)n \rfloor} < \frac{2}{1-\beta}$ rounds of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$.

Thus, the probability that each bin contains at least $(1-\epsilon^2)(1-\beta)c$ honest players in every execution of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ is $p^{\frac{2}{1-\beta}}$. Henceforth,

$$\begin{aligned}\Pr[\mathsf{GOOD}, \mathsf{Unique}] &\geq p^{\frac{2}{1-\beta}}(1-\mathsf{negl}(\lambda)) \\ &\geq \left(1 - \frac{2}{1-\beta}\frac{n}{c}\exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\}\right)(1-\mathsf{negl}(\lambda)) \\ &\geq 1 - \frac{2}{1-\beta}\frac{n}{c}\exp\left\{-\frac{\epsilon^4}{2}(1-\beta)c\right\} - \mathsf{negl}(\lambda) = 1-\delta.\end{aligned}$$

4.2.4 Proof of Lemma 4.9

In $\mathsf{LBin\text{-}V}$, the players choose their bins in Step 2 with their virtual IDs and broadcast the bin choices using $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. By the property of the functionality, in each execution of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, the coalition has to choose their input, i.e., their bin choices, independently from honest players' bin choices. If the coalition chooses to fail a call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, then t players in the coalition will be wiped out, and honest players choose bins with *fresh* randomness in the next call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Therefore, the coalition's strategy S_l of whether to fail the l-th call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in Step 2 depends only on the output of the first l number of calls to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, and the view $\mathsf{view}_K^{\mathsf{comm}}$ of the coalition $\mathcal{K}$ in Step 1. Still, we use $\mathcal{H}$ to denote the set of honest players, where $|\mathcal{H}| = n - \beta n$.

Let L denote the maximum number of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ calls in Step 2. We use Out_l to denote the output of the l-th call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, and S_l to denote the coalition's strategy of whether to fail the l-th call. The support of S_l is $\{\mathsf{fail}, \mathsf{ok}, \perp\}$. If $S_l = \mathsf{ok}$, then for any $l' \geq l$, $S_{l'} = \perp$. Moreover, $\mathsf{Out}_{l'} = \perp$.

For the l-th call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, let $\mathsf{H}_{i,j}$ denote the event that honest player i chooses bin j in that $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ call. Now we proceed to bound

$$\Pr[\mathsf{H}_{i,j} \mid \mathsf{Out}_1, \dots, \mathsf{Out}_\ell, \mathsf{view}_K^{\mathsf{comm}}, S_1, \dots, S_\ell].$$

Since honest players choose their bins independently in different calls to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, and moreover, the coalition's strategy S_l depends only on $\mathsf{Out}_1, \dots, \mathsf{Out}_l$ and $\mathsf{view}_K^{\mathsf{comm}}$, it follows that

$$\Pr\left[\mathsf{H}_{i,j} \mid \mathsf{Out}_1, \dots, \mathsf{Out}_L, \mathsf{view}_K^{\mathsf{comm}}, S_1, \dots, S_L\right] = \Pr\left[H_{i,j} \mid \mathsf{Out}_l, \mathsf{view}_K^{\mathsf{comm}}\right].$$

Next, we show that given the output of each call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, and $\mathsf{view}_K^{\mathsf{comm}}$, the map between the honest virtual ID and the honest players' identity remains hidden from the coalition $\mathcal{K}$. Formally,

Claim 4.11. *Let V_H and V_K be the (unordered) set of virtual IDs chosen by the honest players and the players in the coalition, respectively, that the transcript of Step 1 binds to, and $\mathsf{view}_K^{\mathsf{comm}}$ be the corresponding view of the malicious coalition in Step 1. Then, for any honest player $i \in \mathcal{H}$,*

$$\gamma_i := \Pr\left[\textit{Honest player } i \textit{ chooses } v_i \in V_H \mid V_H, V_K, \mathsf{view}_K^{\mathsf{comm}}\right] \geq \frac{1}{|\mathcal{H}|} - \mathsf{negl}(\lambda).$$

Proof. By a similar argument as in Claim 4.4, by the non-malleability of NMC, for any i, γ_i in Real is computationally indistinguishable from γ_i in Hyb. Therefore, it suffices to bound γ_i in Hyb.

In Hyb, the virtual IDs of the coalition and the view $\mathsf{view}_K^{\mathsf{comm}}$ are independent of honest players' virtual IDs. Hence, in Hyb, for any $i \in \mathcal{H}$,

$$\gamma_i = \Pr\left[\text{Honest player } i \text{ chooses } v_i \in V_H \mid V_H, V_K, \mathsf{view}_K^{\mathsf{comm}}\right] = \frac{1}{|\mathcal{H}|}.$$

The claim thus follows. □

Recall that Out_l is an unordered *set* of all messages $\{(v_i, b_i)\}_{i\in[n]}$, where v_i is the virtual ID chosen by player i and b_i is the bin choice of player i in the l-th call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ (If $|\mathsf{Out}_l| < n$, just pad $n - |\mathsf{Out}_l|$ copies of $(\perp, \perp)$ to Out_l). For $j \in [B]$, we use V_j to denote the set $V_j = \{v_i : (v_i, j) \in \mathsf{Out}_l\}$, i.e., the set of virtual IDs choosing bin j. By Claim 4.11, we have that

$$\begin{aligned}
&\Pr[\mathsf{H}_{i,j} \mid \mathsf{Out}_l = \{(v_i, b_i)\}_{i\in[n]}, \mathsf{view}_K^{\mathsf{comm}} = v] \\
&= \Pr\left[\text{Player } i \text{ chooses virtual ID } v_i \in V_j \,\middle|\, \begin{array}{c} \mathsf{Out}_l = ((v_1, b_1), \dots, (v_n, b_n)), \\ \mathsf{view}_K^{\mathsf{comm}} = v \end{array}\right] \\
&\geq \frac{h_j}{|\mathcal{H}|} - \mathsf{negl}(\lambda),
\end{aligned}$$

where h_j is the number of honest players in bin j. Given the assumption that GOOD happens, $h_j \geq (1-\epsilon^2)(1-\beta)c$ for every $j \in [B]$. Hence, for any output $((v_1, b_1), \dots, (v_n, b_n))$, and any view v,

$$\Pr\left[\mathsf{H}_{i,j} \,\middle|\, \begin{array}{c} \mathsf{Out}_l = ((v_1, b_1), \dots, (v_n, b_n)), \\ \mathsf{view}_K^{\mathsf{comm}} = v, \mathsf{GOOD} \end{array}\right] \geq \frac{(1-\epsilon^2)c}{n} - \mathsf{negl}(\lambda). \tag{4}$$

Note that the lightest bin b^* is a deterministic function given the outputs of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ and the coalition's strategy: given $\mathsf{Out}_1, \ldots, \mathsf{Out}_L$ and $S_1, \ldots, S_L$, b^* is the lightest bin in the first success call to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Let H_i denote the event that honest player i is chosen into the committee.

We use view_K to denote the view of the adversary at the end of Step 2, which includes $\mathsf{view}_K^{\mathsf{comm}}$, and $\mathsf{Out}_1, \ldots, \mathsf{Out}_L$, as well as $\mathcal{A}$'s strategy $S_1, \ldots, S_L$. Then, for any $i \in \mathcal{H}$,

$$\begin{aligned}
&\Pr[\mathsf{H}_i \mid \mathsf{GOOD}] = \sum_{j\in[B]} \Pr\left[\mathsf{H}_{i,j}, b^* = j \mid \mathsf{GOOD}\right] \\
=&\sum_{j\in[B]} \sum_{v\in\mathsf{supp}(\mathsf{view}_K)} \Pr\left[\mathsf{H}_{i,j}, b^* = j \mid \mathsf{view}_K = v, \mathsf{GOOD}\right] \cdot \Pr\left[\mathsf{view}_K = v \mid \mathsf{GOOD}\right] \\
\geq &\sum_{v\in\mathsf{supp}(\mathsf{view}_K)} \left(\frac{(1-\epsilon^2)c}{n} - \mathsf{negl}(\lambda)\right) \Pr\left[\mathsf{view}_K = v \mid \mathsf{GOOD}\right] \\
=&\frac{(1-\epsilon^2)c}{n} - \mathsf{negl}(\lambda),
\end{aligned}$$

where the inequality follows from (4), and the last equality results from the fact that the sum of probability over the whole support is 1.

Therefore, at the end of Step 2, the probability that an honest player i's virtual ID is in the lightest bin b^* is at least $(1-\epsilon^2)c/n - \mathsf{negl}(\lambda)$. As the event Unique happens (Recall that U_{b^*} is the set of virtual IDs that are unique and chooses the lightest bin b^*), if honest player i's virtual ID v_i is in the lightest bin, then v_i is also in the set U_{b^*}. Thus, player i will be elected into the committee. This implies that the honest player i will be elected into the committee with a probability at least $(1-\epsilon^2)c/n - \mathsf{negl}(\lambda) \geq (1-\epsilon)c/n$, given that GOOD and Unique happens.

4.2.5 Proof of Lemma 4.10

The proof to this Lemma is similar to the proof of Lemma 4.2, except that now we do not commit and then open the bin choices. Instead, the players directly send their bin choice with the virtual ID using the anonymous broadcast functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Note that in Step 2, the coalition can choose to fail $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ and change their bin choices. Yet, in the next run of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, honest players choose their bins with fresh randomness, and the coalition's bin choices are still independent of the honest players' bin choices in the new call. Therefore, the proof of Lemma 4.2 still applies in each call of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Consequently, the fraction of the coalition in the lightest bin at the end of Step 2 is at most $\frac{\beta}{1-2\epsilon}$ given that GOOD happens. In addition, since Unique happens, every honest player choosing the lightest bin will appear in the committee. The fraction of the coalition in the committee is thus at most $\frac{\beta}{1-2\epsilon}$ given GOOD and Unique happens. That is, $\mathbb{E}[\widetilde{\beta} \mid \mathsf{GOOD}, \mathsf{Unique}] \leq \frac{\beta}{1-2\epsilon}$. In addition, Corollary 4.6 still holds in $\mathsf{LBin\text{-}V}$.

5 Fairness Amplification Though Iteration

This section gives our final game-theoretically fair committee election and leader election protocols to select arbitrary committee size with good fairness parameters. The committee election protocol $\mathsf{LBin\text{-}V}$ introduced in Section 4.2 does not achieve fairness with good parameter for arbitrary committee size. For example, if we want to choose a $\log\log n$-sized committee from n players using $\mathsf{LBin\text{-}V}$, the probability that the GOOD event does not happen is upper bounded by $\frac{n}{\log\log n}\exp\{-\frac{\epsilon^4}{2}\log\log n\}$, which is even larger than 1. This makes $\mathsf{LBin\text{-}V}$ not fair enough for electing a small sized-committee.

Therefore, to build a fair committee election protocol that works for arbitrary committee size, we compose $\mathsf{LBin\text{-}V}$ for multiple iterations, and combine it with the tournament tree protocol if necessary.

We first give the formal description of the tournament tree protocol and its "almost perfect" fairness. Then we give our final committee election protocol that achieves game-theoretic fairness for arbitrary committee size.

5.1 Preliminary: Fairness of Tournament Tree Protocol

This section gives a formal description of the tournament tree protocol.

Tournament tree protocol $\mathsf{Tourn}(\mathcal{O})$

Let n be the size of $\mathcal{O}$.

- If $n = 1$, return the single player in $\mathcal{O}$.
- Otherwise, let $n_1 = \lfloor \frac{n}{2} \rfloor$ and $n_2 = \lceil \frac{n}{2} \rceil$. Let $\mathcal{O}_1$ be the first n_1 players in $\mathcal{O}$ and $\mathcal{O}_2$ be the remaining players.
- In parallel, run $\mathsf{Tourn}(\mathcal{O}_1)$ and $\mathsf{Tourn}(\mathcal{O}_2)$, and denote the output as O_1 and O_2, respectively.
- The final winner is determined by the duel protocol between O_1 and O_2 such that O_i wins with probability n_i/n. This is described below.

Duel Protocol between O_1 and O_2

Let $\frac{k_1}{k_1+k_2}$ and $\frac{k_2}{k_1+k_2}$ be the probability that player O_1 and O_2 wins, respectively.

- Let $k = k_1 + k_2$, and $\ell = \lceil \log k \rceil$. Each player O_i commits to an ℓ-bit random string that represents some $s_i \in \mathbb{Z}_{k-1}$ for $i = 1, 2$.
- Each player O_i opens its commitment and reveals s_i. If $s_1 + s_2 \mod k \in \{0, \ldots, k_1 - 1\}$, player O_1 wins. Otherwise, O_2 wins.
- If a player aborts or fails to open the commitment correctly, it is treated as forfeiting and the other player wins.

Lemma 5.1 (Theorem 3.5 of Chung et al. [CCWS21]). *Let n be the number of players and λ be the security parameter. Then, the tournament-tree protocol, when instantiated with a suitable publicly verifiable, non-malleable commitment scheme as defined in Section 3.3, satisfies $(1 - \mathsf{negl}(\lambda))$-CSP-fairness and $(1 - \mathsf{negl}(\lambda))$-maximin-fairness against coalition of arbitrarily sizes. Moreover, the round complexity is $O(\log n)$.*

5.2 Our Final Game-Theoretically Fair Committee Election

In this section, we give our fair committee election protocol that works for arbitrary committee size. Our final protocol runs multiple iterations of $\mathsf{LBin\text{-}V}$ and combines it with the tournament tree protocol if necessary. The $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ ideal functionality in $\mathsf{LBin\text{-}V}$ can be instantiated in real-world cryptography, with only a constant round blowup. The instantiation will be given in Section 6.

Let c be the upper bound of the committee size we want to achieve. The final committee election is given below.

Committee election protocol $\mathsf{CElect}(n, c)$

Parameter: Let c be the upper bound of the committee size and R be the round complexity we want to achieve. The initial committee is $\mathcal{C}_0 = [n]$, $c_0 = n$. The fraction of the coalition is $\beta_0 = \beta$. If $c \geq 2^R$, let $L \leq R$ be the smallest integer such that $\log^{(L)} n \leq c^{0.1}$ and $\epsilon = \frac{1}{c^{0.1}}$; otherwise, set $L \leq R$ be the smallest integer such that $\log^{(L)} n \leq 2^R$ and $\epsilon = \frac{1}{2^R}$.

Protocol

1. For $\ell = 1, \dots, L-1$:
   - Let $c_\ell = (\log^{(\ell)} n)^{10}$, $\mathcal{O} = \mathcal{C}_{\ell-1}$, $\beta_\ell = \beta_{\ell-1}(1-\epsilon^2) + \epsilon^2$.
   - Run $\mathsf{LBin\text{-}V}(c_\ell, \mathcal{C}_{\ell-1}, \beta_{\ell-1})$. That is, we choose a committee $\mathcal{C}_\ell$ of size $c_\ell = (\log^{(\ell)} n)^{10}$ from $\mathcal{C}_{\ell-1}$.
   - $\ell = \ell + 1$.
2. If $c \geq 2^R$, set $c_L = c$; otherwise, set $c_L = 2^{11R}$. Run the committee election protocol $\mathsf{LBin\text{-}V}(c_L, \mathcal{C}_{L-1}, \beta_{L-1})$ to elect a committee $\mathcal{C}_L$ of size at most c_L.
3. If $c_L \geq c$, run c number of parallel instances of $\mathsf{Tourn}^{sid}(\mathcal{C}_L)$ for $sid \in [c]$. Let the final committee be the set of elected leaders in these c instances of tournament tree protocol.

Note that in the protocol, β_ℓ is just a parameter that passes to $\mathsf{LBin\text{-}V}$, together with c and $\mathcal{O}$. It is *not* the real fraction of the coalition in committee $\mathcal{C}_\ell$. Instead, it is the upper bound of the real fraction of the coalition in $\mathcal{C}_\ell$ if *good* event happens in each round up to ℓ. The parameter β_ℓ is only used to set the parameter t of $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in the ℓ-th $\mathsf{LBin\text{-}V}$ call.

Theorem 5.2. *Assume the existence of enhanced trapdoor permutations and collision-resistant hash functions. Fix n and c. Let L^* be the smallest integer such that $\log^{(L^*)} n \leq c$. Then for any $L^* \leq R \leq C_0 \log n$ for some constant C_0, we have that*

- *If $c \geq 2^R$, there exists an $O(R)$-round committee election that achieves $(1 - \frac{1}{c^{\Theta(1)}})$-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most $(1 - \frac{L^*}{\Theta(R)})n$.*
- *If $c < 2^R$, there exists an $O(R)$-round committee election that achieves $(1 - \frac{1}{2^{\Theta(R)}})$-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most $(1 - \frac{L}{\Theta(R)})n$, where L is the smallest integer such that $\log^{(L)} n \leq 2^R$.*

Proof. Consider the CElect protocol given above. By the theorem statement, L is the number of invocations of the committee election protocol $\mathsf{LBin\text{-}V}$. The size of the committee chosen in each round of $\mathsf{LBin\text{-}V}$ satisfies that $c_\ell = (\log^{(\ell)} n)^{10}$ for $\ell = 1, \dots, L-1$ and $c_L = c$ if $c \geq 2^R$ and $c_L = 2^{11R}$ otherwise. We use $\widetilde{\beta_\ell}$ to denote the random variable of the fraction of the coalition in committee $\mathcal{C}_\ell$ and $\mathcal{B}_\ell$ to denote the support of β_ℓ.

If $\mathbf{c \geq 2^R}$: In this case, the protocol runs L rounds of $\mathsf{LBin\text{-}V}$. Recall that L^* is the smallest integer such that $\log^{(L^*)} n \leq c$, we have that $L \leq L^* \leq L + C^*$ for some non-negative constant C^*. Therefore, $\beta \leq 1 - \frac{L^*}{\Theta(R)} \leq 1 - \frac{L}{\Theta(R)}$.

Round complexity: By Theorem 4.7, the round complexity of the ℓ-th committee election $\mathsf{LBin\text{-}V}$

is at most $\frac{2C'}{1-\beta_\ell}+2$ for a constant C'. Since $\beta_\ell = \beta_{\ell-1}(1-\epsilon^2)+\epsilon^2$, we have that, for any $\ell \in [L]$,

$$\begin{aligned}
\beta_\ell &= \beta_{\ell-1}(1-\epsilon^2)+\epsilon^2 \\
&= \beta(1-\epsilon^2)^\ell + \epsilon^2 \sum_{i=0}^{\ell-1}(1-\epsilon^2)^i \\
&= \beta(1-\epsilon^2)^\ell + \epsilon^2 \cdot \frac{1-(1-\epsilon^2)^\ell}{\epsilon^2} \\
&= 1-(1-\beta)(1-\epsilon^2)^\ell \\
&\le 1-\frac{L}{\Theta(R)}\left(1-\frac{\ell}{c^{0.1}}\right) \\
&\le 1-\frac{L}{\Theta(R)}\left(1-\frac{\ell}{2^{\Theta(R)}}\right),
\end{aligned} \tag{5}$$

where the last inequality comes from the assumption that $c \ge 2^R$. Therefore, the round complexity of the committee election CElect protocol is

$$\begin{aligned}
&\sum_{\ell=1}^{L}\left(\frac{2C'}{1-\beta_\ell}+2\right) \\
\le& 2L + L\cdot\frac{2C}{1-\beta_L} \\
\le& L\cdot\frac{2C'}{\frac{L}{\Theta(R)}\left(1-\frac{L}{2^{\Theta(R)}}\right)} + O(R) && \text{By } L \le R \text{ and (5)} \\
=& \frac{C'\cdot\Theta(R)}{1-\frac{L}{2^{\Theta(R)}}} + O(R) = O(R).
\end{aligned}$$

We now proceed to prove the CSP fairness and the maximin fairness separately.

CSP fairness: By Theorem 4.7, the $\mathsf{LBin\text{-}V}$ in the ℓ-th round is $(1-2\epsilon, \delta)$-CSP fair against a coalition of size $\widetilde{\beta}_{\ell-1}c_{\ell-1}$, where

$$\begin{aligned}
\delta_\ell &= \frac{2}{1-\beta_\ell}\frac{c_{\ell-1}}{c_\ell}\exp\left\{-\frac{\epsilon^4}{2}(1-\widetilde{\beta}_{\ell-1})c_\ell\right\} + \mathsf{negl}(\lambda) \\
&\le (\log^{(\ell-1)} n)^{10}\exp\left\{-5(1-\widetilde{\beta}_{\ell-1})(\log^{(\ell)} n)^6\right\}, && \text{for } \ell = 1,\dots,L-1 \\
\delta_L &= \frac{2}{1-\beta_L}\frac{c_{L-1}}{c}\exp\left\{-\frac{\epsilon^4}{2}(1-\widetilde{\beta}_{L-1})c\right\} + \mathsf{negl}(\lambda) \\
&\le (\log^{(L-1)} n)^{10}\exp\left\{-5(1-\widetilde{\beta}_{L-1})c^{0.6}\right\}.
\end{aligned}$$

By definition, for $\ell = 1,\dots,L$, there exists an event G_ℓ that satisfies the following:

$$\Pr[\mathsf{G}_\ell \mid \widetilde{\beta}_{\ell-1} = x] \ge 1-\delta_\ell(x), \tag{6}$$

$$\mathbb{E}[\widetilde{\beta}_\ell \mid \mathsf{G}_\ell, \widetilde{\beta}_{\ell-1} = x] \le \frac{x}{1-2\epsilon}, \tag{7}$$

where

$$\begin{aligned}
\delta_\ell(x) &= (\log^{(\ell-1)} n)^{10}\exp\left\{-5(1-x)(\log^{(\ell)} n)^6\right\}, && \text{for } \ell = 1,\dots,L-1 \\
\delta_L(x) &= (\log^{(L-1)} n)^{10}\exp\left\{-5(1-x)c^{0.6}\right\}.
\end{aligned}$$

Let G denote the event that $\mathsf{G}_1, \ldots, \mathsf{G}_L$ happens.

We first bound $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}]$. For $\ell = 1, \ldots, L$, we have that

$$\begin{aligned}
&\mathbb{E}[\widetilde{\beta}_\ell \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] \\
=& \sum_{x \in \mathcal{B}_{\ell-1}} \mathbb{E}\left[\widetilde{\beta}_\ell \,\middle|\, \begin{array}{c} \widetilde{\beta}_{\ell-1} = x, \\ \mathsf{G}_1, \ldots, \mathsf{G}_\ell \end{array}\right] \Pr[\widetilde{\beta}_{\ell-1} = x \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] \\
\leq& \sum_{x \in \mathcal{B}_{\ell-1}} \frac{x}{1-2\epsilon} \Pr[\widetilde{\beta}_{\ell-1} = x \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] && \text{(By (7))} \\
=& \frac{1}{1-2\epsilon} \mathbb{E}[\widetilde{\beta}_{\ell-1} \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}]
\end{aligned}$$

Consequently $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}] = \mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_L] \leq \frac{\beta}{(1-2\epsilon)^L}$.

To compute the probability that G happens, note that for $\ell = 1, \ldots, L$

$$\begin{aligned}
&\Pr[\overline{\mathsf{G}_\ell} \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}] \\
=& \sum_{x \in \mathcal{B}_{\ell-1}} \Pr\left[\overline{\mathsf{G}_\ell} \,\middle|\, \begin{array}{c} \widetilde{\beta}_{\ell-1} = x, \\ \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1} \end{array}\right] \Pr[\widetilde{\beta}_{\ell-1} = x \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}] \\
\leq& \sum_{x \in \mathcal{B}_{\ell-1}} \delta_\ell(x) \Pr[\widetilde{\beta}_{\ell-1} = x \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}] \\
=& \mathbb{E}[\delta_\ell(\widetilde{\beta}_{\ell-1}) \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}].
\end{aligned}$$

We give the following claim and show that the result follows from it. The proof of the claim is given afterward.

Claim 5.3. *There exists a constant C, such that or any $\ell = 1, \ldots, L$, the expectation*

$$\mathbb{E}[\delta_\ell(\widetilde{\beta}_{\ell-1}) \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}] \leq \exp\left\{-C(\log^{(\ell)} n)^4\right\}.$$

Therefore, the probability that G happens is at least

$$\begin{aligned}
\Pr[\mathsf{G}] &= \prod_{\ell=1}^{L} \Pr[\mathsf{G}_\ell \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell-1}] \\
&\geq \prod_{\ell=1}^{L} \left(1 - \exp\left\{-C(\log^{(\ell)} n)^4\right\}\right) \geq 1 - L\exp\left\{-C(\log^{(L)} n)^4\right\}.
\end{aligned}$$

Thus, CElect is a $(1-(1-2\epsilon)^L, \delta)$-CSP fair (c_L, n)-committee election, where the probability that G does not happen is at most $\delta = L\exp\left\{-C(\log^{(L)} n)^4\right\}$. By Lemma 3.8, CElect is a $(1-\epsilon_1)$-CSP fair leader election, where

$$\epsilon_1 = \frac{\beta(1-(1-2\epsilon)^L) + \delta(1-2\epsilon)^L}{\beta + \delta(1-2\epsilon)^L} + \mathsf{negl}(\lambda) \leq \frac{\beta \cdot 2L\epsilon + \delta}{\beta + \delta}, \tag{8}$$

where the inequality comes from the fact that $(1-2\epsilon)^L \geq 1 - 2L\epsilon$, and that for any $0 < \frac{a}{b} < 1$, $\frac{a}{b} \leq \frac{a+\varepsilon}{b+\varepsilon}$ for $\varepsilon > 0$. We consider the following cases based on the fraction of the coalition β.

- Case 1: $\beta \geq \frac{\delta}{\epsilon}$. Substitute into (8), we have
$$\epsilon_1 \leq \frac{\beta \cdot 2L\epsilon + \delta}{\beta + \delta} \leq \frac{\beta \cdot 2L\epsilon + \beta\epsilon}{\beta + \beta\epsilon} = \frac{(2L+1)\epsilon}{1+\epsilon} = \frac{1}{c^{\Theta(1)}}.$$
Thus, the committee election protocol CElect is $\left(1 - \frac{1}{c^{\Theta(1)}}\right)$-CSP fair.

- Case 2: $\beta < \frac{\delta}{\epsilon} = \frac{L}{\epsilon \exp\{C(\log^{(L)} n)^4\}}$. In this case, substituting β into (8) directly does not yield the desired result. However, note that the fraction of the coalition is very small. With some large probability, there will be no coalition's representation on the committee in the last few rounds of committee election LBin-V, and the coalition gains utility 0 no matter the good events happen or not in the last few rounds.

  Let $\ell^* \in [L]$ be the value such that $\frac{L}{\epsilon \exp\{C(\log^{(\ell^*-1)} n)^4\}} \leq \beta < \frac{L}{\epsilon \exp\{C(\log^{(\ell^*)} n)^4\}}$. Note that since $\beta \geq \frac{1}{n}$ (otherwise we are in an all-honest execution), such ℓ^* must exist.

  Since the expected fraction of the coalition in $\mathcal{C}_\ell$ is $\mathbb{E}[\widetilde{\beta}_\ell \mid \mathsf{G}_1 \ldots \mathsf{G}_\ell] \leq \frac{\beta}{(1-2\epsilon)^\ell}$, the expected number of the coalition's representations f_ℓ in $\mathcal{C}_\ell$ is $\mathbb{E}[f_\ell \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] \leq \frac{\beta c_\ell}{(1-2\epsilon)^\ell}$. For any $\ell \geq \ell^* - 1$, the expected number of the coalition's representation in the committee is
$$\mathbb{E}[f_\ell \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] \leq \frac{\beta c_\ell}{(1-2\epsilon)^\ell} < \frac{L \cdot 2^R (\log^{(\ell)} n)^{10}}{\exp\{C(\log^{(\ell^*)} n)^4\}(1-2\epsilon)^\ell} < 1.$$
  If there is no coalition's representation in $\mathcal{C}_\ell$ for some $\ell \in [L]$, the expected utility of the coalition is 0, no matter whether the good events happen or not in the later committee elections, i.e., for any $\ell' > \ell$, $\mathbb{E}[\widetilde{\beta}_{\ell'} \mid f_\ell < 1] = 0$.

  By Markov inequality, for any $\ell \geq \ell^* - 1$, the probability that there exist players from the coalition in the ℓ-th committee is $\mathcal{C}_\ell$ is
$$\Pr\left[f_\ell \geq 1 \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell\right] \leq \frac{\beta c_\ell}{(1-2\epsilon)^\ell}. \tag{9}$$
  Let G' denote the event that $\mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*}$ happens. Then the probability that G' happens is at least $(\ell^* - 1)\exp\{-C(\log^{(\ell^*-1)} n)^4\}$ by a similar argument as before. Next we proceed to bound $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}']$. Recall that f_ℓ is the number of the coalition's representations in $\mathcal{C}_\ell$, we have
$$\begin{aligned}
&\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}] \\
&= \sum_{\ell=\ell^*-1}^{L} \mathbb{E}\left[\widetilde{\beta}_L \;\middle|\; \begin{matrix} \mathsf{G}_1, \ldots, \mathsf{G}_\ell, \\ f_{\ell-1} \geq 1, f_\ell < 1 \end{matrix}\right] \Pr\left[\begin{matrix} \mathsf{G}_{\ell^*}, \ldots, \mathsf{G}_\ell, \\ f_{\ell-1} \geq 1, f_\ell < 1 \end{matrix} \;\middle|\; \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}\right] \\
&+ \sum_{\ell=\ell^*-1}^{L-1} \mathbb{E}\left[\widetilde{\beta}_L \;\middle|\; \begin{matrix} \mathsf{G}_1, \ldots, \mathsf{G}_\ell, \\ f_\ell \geq 1, \overline{\mathsf{G}_{\ell+1}} \end{matrix}\right] \Pr\left[\begin{matrix} \mathsf{G}_{\ell^*}, \ldots, \mathsf{G}_\ell, \\ f_\ell \geq 1, \overline{\mathsf{G}_{\ell+1}} \end{matrix} \;\middle|\; \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}\right] \\
&+ \mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_L, f_L \geq 1] \Pr[\mathsf{G}_{\ell^*}, \ldots, \mathsf{G}_L, f_L \geq 1 \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}].
\end{aligned} \tag{10}$$
  Intuitively, the first term is the expectation of $\widetilde{\beta}_L$ if the number of the coalition's representation after ℓ rounds of LBin-V becomes 0; the second term is the expectation of $\widetilde{\beta}_L$ given that there exist the coalition's representations in each round, yet the good event does not happen in the ℓ-th round of LBin-V; the third term is the expected fraction of the coalition given that good event happens in every round of LBin-V, and that the number of the coalition's representations in every round is at least one. We now proceed to calculate the above terms separately.

1. For any $\ell \geq \ell^* - 1$, $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell, f_{\ell-1} \geq 1, f_\ell < 1] = 0$.
2. For any $L - 1 \geq \ell \geq \ell^* - 1$, $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell, f_{\ell-1} \geq 1, \overline{\mathsf{G}_{\ell+1}}] \leq 1$. The probability
$$\begin{aligned} &\Pr[\mathsf{G}_{\ell^*}, \ldots, \mathsf{G}_\ell, f_\ell \geq 1, \overline{\mathsf{G}_{\ell+1}} \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}] \\ =&\Pr[\overline{\mathsf{G}_{\ell+1}} \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell, f_\ell \geq 1] \\ &\cdot \Pr[f_\ell \geq 1 \mid \mathsf{G}_1, \ldots, \mathsf{G}_\ell] \Pr[\mathsf{G}_{\ell^*}, \ldots, \mathsf{G}_\ell \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}] \\ \leq& \exp\{-C(\log^{(\ell+1)} n)^4\} \frac{\beta c_\ell}{(1-2\epsilon)^\ell} \leq \frac{\beta}{\exp\{C'(\log^{(\ell+1)} n)^2\}}, \end{aligned}$$
for some constant C'.
3. $\mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_L, f_L \geq 1] \leq \frac{\beta}{(1-2\epsilon)^L}$.

Substituting all three terms back into (10):
$$\begin{aligned} \mathbb{E}[\widetilde{\beta}_L \mid \mathsf{G}_1, \ldots, \mathsf{G}_{\ell^*-1}] &\leq \frac{\beta}{(1-2\epsilon)^L} + \sum_{\ell=\ell^*-1}^{L-1} \frac{\beta}{\exp\{C'(\log^{(\ell+1)} n)^2\}} \\ &\leq \beta \left[\frac{1}{(1-2\epsilon)^L} + L \exp\{-C'(\log^{(L)} n)^2\}\right] \leq \beta \left[\frac{1}{(1-2\epsilon)^L} + \Theta(\epsilon)\right]. \end{aligned}$$

Therefore, for $\frac{L}{\epsilon \exp\{C'(\log^{(\ell^*-1)} n)^4\}} \leq \beta < \frac{L}{\epsilon \exp\{C'(\log^{(\ell^*)} n)^4\}}$, the expected fraction of the coalition in the committee is at most $\beta \left[\frac{1}{(1-2\epsilon)^L} + \Theta(\epsilon)\right] + \delta'$, where
$$\delta' = (\ell^* - 1) \exp\{-C'(\log^{(\ell^*-1)} n)^4\} \leq \beta\epsilon.$$

That is, the expected fraction of the coalition in the committee is at most
$$\beta \left[\frac{1}{(1-2\epsilon)^L} + \Theta(\epsilon)\right] \leq \frac{\beta}{1 - \frac{1}{c^{\Theta(1)}}}.$$

Therefore, the committee election protocol is $(1-\epsilon_1)$-CSP fair for $\epsilon_1 = \frac{1}{c^{\Theta(1)}}$.

Maximin fairness: Let $\mathsf{H}_{i,\ell}$ be the event that an honest individual i gets elected into the committee $\mathcal{C}_\ell$. Then by Theorem 4.7, for $\ell = 1, \ldots, L$,
$$\Pr[\mathsf{H}_{i,\ell} \mid \mathsf{H}_{i,\ell-1}, \mathsf{G}_\ell] \geq \frac{(1-\epsilon)c_\ell}{c_{\ell-1}}, \qquad \Pr\left[\mathsf{H}_{i,\ell} \mid \overline{\mathsf{H}_{i,\ell-1}}\right] = 0.$$

Still, let G denote the event that $\mathsf{G}_1, \ldots, \mathsf{G}_L$ happens. Then the probability that G happens is at least $\Pr[\mathsf{G}] \geq 1 - L \exp\left\{-C(\log^{(L)} n)^4\right\}$. Therefore, we have
$$\begin{aligned} \Pr[\mathsf{H}_{i,L} \mid \mathsf{G}] &\geq \Pr[\mathsf{H}_{i,L} \mid \mathsf{H}_{i,L-1}, \mathsf{G}_1, \ldots, \mathsf{G}_L] \Pr[\mathsf{H}_{i,L-1} \mid \mathsf{G}_1, \ldots, \mathsf{G}_L] \\ &\geq \frac{(1-\epsilon)c_L}{c_{L-1}} \Pr[\mathsf{H}_{i,L-1} \mid \mathsf{G}_1, \ldots, \mathsf{G}_{L-1}] \Pr[\mathsf{G}_1, \ldots, \mathsf{G}_L] \\ &\geq \ldots \\ &\geq \frac{c_L}{n}(1-\epsilon)^L \left(1 - L \exp\left\{-C(\log^{(L)} n)^4\right\}\right) \\ &\geq \frac{c_L}{n}\left(1 - L\epsilon - L \exp\left\{-C(\log^{(L)} n)^4\right\}\right) \geq \frac{c_L}{n}(1 - 2L\epsilon) \end{aligned}$$

Thus, CElect is a $(1-2L\epsilon, \delta)$-maximin fair committee election, where $\delta \leq L \exp\left\{-C(\log^{(L)} n)^4\right\}$. It follows from Lemma 3.9 that the committee election CElect is $(1 - \frac{1}{c^{\Theta(1)}})$-maximin fair.

If $\mathbf{c < 2^{10R}}$ In this case, the protocol runs L rounds of $\mathsf{LBin\text{-}V}$ and then c parallel instance of the tournament tree protocol, which takes at most $O(R)$ rounds. Pick $\epsilon = \frac{1}{2^{\Theta(R)}}$, by a similar argument as above, we have that the expected fraction of the coalition in $\mathcal{C}_L$ (the committee chosen by the L-th $\mathsf{LBin\text{-}V}$) is at most $\frac{\beta}{1-\epsilon}$, and the probability that an honest individual i gets elected into $\mathcal{C}_L$ is at least $\frac{(1-\epsilon)c_L}{n}$, where $c_L = |\mathcal{C}_L|$.

<u>CSP fairness</u>: Let X_i be an indicator of whether the elected leader in the k-th instance of $\mathsf{Tourn}(\mathcal{C}_L)$ is a player from the coalition. Then by Lemma 5.1, $\mathbb{E}[X_k] \le \frac{\beta}{1-\epsilon} + \mathsf{negl}(\lambda)$. Therefore, the expected fraction of the coalition in the final committee is

$$\mathbb{E}\left[\frac{\sum_{k\in[c]} X_k}{c}\right] = \frac{1}{c}\sum_{i\in[c]} \mathbb{E}[X_k] \le \frac{\beta}{(1-\epsilon)(1-\mathsf{negl}(\lambda))} \le \frac{\beta}{1-\frac{1}{2^{\Theta(R)}}}.$$

Therefore, CElect is a $(1-\frac{1}{2^{\Theta(R)}})$-CSP fair committee election.

<u>Maximin fairness</u>: For any fixed honest individual i, let Y_k be an indicator of whether honest player i gets elected as the leader of the k-th instance of $\mathsf{Tourn}(\mathcal{C}_L)$. Then by Lemma 5.1, we have that $\Pr[Y_k \mid \text{Player } i \text{ in } \mathcal{C}_L] \ge \frac{1-\mathsf{negl}(\lambda)}{c_L}$. Therefore,

$$\begin{aligned}
&\Pr[\text{Player } i \text{ in the committee}]\\
=&\Pr[\text{Player } i \text{ in the committee} \mid \text{Player } i \text{ in } \mathcal{C}_L]\Pr[\text{Player } i \text{ in } \mathcal{C}_L]\\
=&\left(1-\prod_{k\in[c]}(1-\Pr[Y_k \mid \text{Player } i \text{ in } \mathcal{C}_L])\right)\frac{(1-\epsilon)c_L}{n}\\
\ge&\left(1-\prod_{k\in[c]}\frac{1-\mathsf{negl}(\lambda)}{c_L}\right)\frac{(1-\epsilon)c_L}{n}\\
\ge&\left(\frac{c}{c_L}-\frac{c(c-1)}{2c_L^2}-\mathsf{negl}(\lambda)\right)\frac{(1-\epsilon)c_L}{n} \ge \frac{c}{c_L}\left(1-\frac{1}{2^R}\right)\frac{(1-\epsilon)c_L}{n}\\
\ge&\frac{c}{n}(1-\frac{1}{2^{\Theta(R)}}).
\end{aligned}$$

Therefore, CElect is $(1-\frac{1}{2^{\Theta(R)}})$-maximin fair against at most $(1-\frac{L}{\Theta(R)})n$. The theorem thus follows. □

Proof of Claim 5.3. Recall that $\beta_\ell = \beta_{\ell-1}(1-\epsilon^2)+\epsilon^2$ for $\ell = 1,\ldots,L$. By Corollary 4.6, the probability that $\Pr[\widetilde{\beta}_\ell \le x(1-\epsilon^2)+\epsilon^2 \mid \widetilde{\beta}_{\ell-1} = x, \mathsf{G}_\ell] = 1$. Therefore, given that $\mathsf{G}_1,\ldots,\mathsf{G}_\ell$ happens, $\widetilde{\beta}_\ell \le \beta_\ell$ with probability 1.

Now we proceed to bound $\mathbb{E}[\delta_\ell(\widetilde{\beta}_{\ell-1}) \mid \mathsf{G}_1,\ldots,\mathsf{G}_{\ell-1}]$. By the definition of expectation and the fact that $\delta_\ell(x)$ is monotone, we have that

$$\begin{aligned}
&\mathbb{E}[\delta_\ell(\widetilde{\beta}_{\ell-1}) \mid \mathsf{G}_1,\ldots,\mathsf{G}_{\ell-1}] = \sum_{x\in\mathcal{B}_{\ell-1}} \delta_\ell(x)\Pr[\beta_{\ell-1} = x]\\
&= \sum_{x\le\beta_{\ell-1}} \delta_\ell(x)\Pr[\widetilde{\beta}_{\ell-1} = x \mid \mathsf{G}_1,\ldots,\mathsf{G}_{\ell-1}]\\
&\le \delta_\ell(\beta_{\ell-1})
\end{aligned}$$

For $\ell = 1, \dots, L-1$, we have that

$$\begin{aligned}\delta_\ell(\beta_{\ell-1}) &= (\log^{(\ell-1)} n)^{10} \exp\left\{-\frac{5L}{\Theta(R)}\left(1-\frac{\ell}{2^R}\right)(\log^{(\ell)} n)^6\right\} \\ &\leq \exp\{-C(\log^{(\ell)} n)^4\}.\end{aligned}$$

For $\ell = L$, we have that

$$\begin{aligned}\delta_L(\beta_{L-1}) &\leq (\log^{(L-1)} n)^{10} \exp\left\{-\frac{5L}{\Theta(R)}\left(1-\frac{L}{c^{0.1}}\right)c^{0.6}\right\} \\ &\leq \exp\{-C(\log^{(L)} n)^4\},\end{aligned}$$

where the last inequality follows from the fact that $\log^{(L)} n \leq c^{0.1}$. The claim thus follows. □

Remark 5.4. *As we will see in Theorem 6.2, the parameter β_ℓ should be appropriately chosen such that in each round, $c_\ell - t \geq \widetilde{\beta}_\ell c_\ell$ where $t = \lfloor (1-\beta_\ell)c_\ell \rfloor$. Note that in the proof, we only care about the case when good events G_ℓ happen in each round. Otherwise, we just assume that the committee contains no honest players, and we do not care about the security of $\mathcal{F}^{t,\mathcal{O}}_{\text{anon}}$. When good events happen in each round, as analyzed in the proof, $\widetilde{\beta}_\ell \leq \beta_\ell$ with probability 1 for $\ell = 1, \dots, L$. Therefore, we have the guarantee that $c_\ell - t \geq \widetilde{\beta}_\ell c_\ell$, as shown in (5).*

Our final leader election protocol can be gained directly by picking $c = 1$ in Theorem 5.2.

Theorem 5.5. *Assume the existence of enhanced trapdoor permutations, and collision-resistant hash functions. Fix n and let $\log^* n \leq R \leq C \log n$ be the round complexity we want to achieve for some constant C. Then there exists an $O(R)$-round leader election that achieves $(1 - \frac{1}{2^{\Theta(R)}})$-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most $(1 - \frac{L}{\Theta(R)})n$, where L is the smallest integer such that $\log^{(L)} n \leq 2^R$.*

Proof. By picking $c = 1$ in Theorem 5.2, we have that $L^* = \log^* n$. Thus, by Theorem 5.2, for $\log^* n \leq R \leq C \log n$, there is an $O(R)$-round committee election that achieves $(1 - \frac{1}{2^{\Theta(R)}})$-game-theoretic fairness against a non-uniform p.p.t. coalition of size at most $(1 - \frac{L}{\Theta(R)})n$, where L is the smallest integer such that $\log^{(L)} n \leq 2^R$. This size-1 committee will be the elected leader. □

6 Instantiation of the Ideal Functionalities

In this section, we show how to instantiate the ideal functionalities $\mathcal{F}^{t,\mathcal{O}}_{\text{anon}}$ used in committee election LBin-V. Recall that the ideal functionality $\mathcal{F}^{t,\mathcal{O}}_{\text{anon}}$ receives one message from each player and either sends the set of all messages it receives to everyone or a set of corrupt players of size at least t to everyone. We will first give our protocol in a IdealZK*-hybrid model in which players have access to an ideal zero-knowledge proof functionality. Then we use the elegant techniques of Pass [Pas04] to instantiate the protocol with real-world cryptography. Next. we will first describe the IdealZK* functionality in Section 6.1, and then we will give our protocol in the IdealZK*-hybrid world in Section 6.2.

6.1 Ideal Zero-Knowledge Functionality IdealZK*

Basically, IdealZK* either sends success to everyone indicating that the proof is correct, or the identity of the prover/verifier who leads to the failure of the proof. Formally,

Ideal Zero-knowledge Functionality $\mathsf{IdealZK}^*[x, L, i, j]$

The functionality involves n parties $1, \ldots, n$, and is parametrized with a statement x, the language L, the prover's identity i and the verifier's identity j.

1. If both the prover i and the verifier j are corrupted, receive a bit b from the prover i. If $b = 1$, send $(\mathsf{success}, i, j)$ to everyone.
2. Receive ok or $\perp$ from the verifier j.
3. If received $\perp$ from the verifier, send (fail, j) to everyone.
4. Receive w or $\perp$ from the prover.
5. If $\mathcal{R}(x, w) = 1$, send $(\mathsf{success}, i, j)$ to everyone. Otherwise send (fail, i) to everyone.

In an n-party $\mathsf{IdealZK}^*$-hybrid protocol, the players can invoke the ideal zero-knowledge functionality $\mathsf{IdealZK}^*[x, L, i, j]$ between any prover $i \in [n]$ and any verifier $j \in [n]$, and for arbitrary NP language L. Without loss of generality, in every round, there can be at most n^2 concurrent invocations of $\mathsf{IdealZK}^*$. Given an n-party $\mathsf{IdealZK}^*$-hybrid protocol, we can instantiate $\mathsf{IdealZK}^*$ with actual cryptography using the elegant techniques suggested by Pass [Pas04].

Theorem 6.1. *(Constant-round, bounded concurrent secure computation [Pas04]). Assume the existence of enhanced trapdoor permutations and collision-resistant hash functions. Then, given an n-party $\mathsf{IdealZK}^*$-hybrid protocol Π^*, in which the number of concurrent calls of $\mathsf{IdealZK}^*$ is upper bounded by a priori known bound $m = \mathsf{poly}(\lambda)$, there exists a real-world protocol Π such that the following hold:*

- **Simulatability***: For every real-world non-uniform p.p.t. adversary $\mathcal{A}$ controlling an arbitrary subset of up to $n-1$ players in Π, there exists a non-uniform p.p.t. adversary $\mathcal{A}^*$ in the protocol Π^*, such that for any input $(x_1, \ldots, x_n)$, every auxiliary string $z \in \{0,1\}^*$,*

  $$\mathsf{Exec}^{\Pi,\mathcal{A}}(1^\lambda, x_1, \ldots, x_n, z) \equiv_c \mathsf{Exec}^{\Pi^*,\mathcal{A}^*}(1^\lambda, x_1, \ldots, x_n, z).$$

  In the above, the notation $\mathsf{Exec}^{\Pi,\mathcal{A}}$ (or $\mathsf{Exec}^{\Pi^*,\mathcal{A}^*}$) outputs each honest players' outputs as well as the corrupt players' (arbitrary) outputs.

- **Round efficiency***: The round complexity of Π is at most a constant factor worse than that of Π^*.*

This real-world protocol is fulfilled by replacing the $\mathsf{IdealZK}^*$ instance with the bounded concurrent zero-knowledge proofs. All the zero-knowledge proof messages are posted to the broadcast channel. We defer the proof of Theorem 6.1 to Appendix A.

Now it suffices to show how to replace $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ with a protocol $\mathsf{Anon}^{t,\mathcal{O}}$ in the $\mathsf{IdealZK}^*$-hybrid world. In the protocol, we will omit the language L when it is clear from the context.

6.2 Implementing Anonymous Broadcast Functionality

In this section, we describe how to implement $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in the $\mathsf{IdealZK}^*$-hybrid model. The protocol makes use of a perfect binding, statistically hiding commitment scheme comm. Also, every player keeps track of two sets, $\mathcal{D}_s$ and $\mathcal{D}_r$, the set of players who fail to share and the set of players who fail to reconstruct, respectively, to guarantee the identifiable abort property. Still, we use $\mathcal{K}$ to represent the set of corrupted players, $\mathcal{H}$ to represent the set of honest players. The number of parallel sessions is set to be λ. The protocol $\mathsf{Anon}^{t,\mathcal{O}}$ is given below.

$\mathsf{Anon}^{t,\mathcal{O}}$: instantiating $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in the $\mathsf{IdealZK}^*$ -hybrid world

Parameters: Let $M = 2n$ be the number of slots. Let $\mathcal{D}_s$, $\mathcal{D}_r$ and Out be initially empty sets. Without loss of generality we assume that $\mathcal{O} = [n]$.

Building blocks: A perfectly binding, computationally hiding commitment scheme comm.

Input: Each player has an input $m_i \in \mathbb{F}$ for a finite field $\mathbb{F}$ with size larger than 2^λ. The sum of tuples is computed entry-wise, i.e., $(a_1, b_1, c_1) + (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$.

Preparation Phase Run the following for λ independent, parallel sessions:

1. Player i uniformly randomly choose a nonce $\mathsf{mid}_i \in \mathbb{F}$. It then uniformly randomly chooses a slot $l_i \leftarrow [M]$ and computes a vector $\mathbf{S}_i \in (\mathbb{F}^3)^M$ such that $\mathbf{S}_i[l] = (0, 0, 0)$ if $l \neq l_i$, and $\mathbf{S}_i[l] = (m_i, \mathsf{mid}_i, 1)$ if $l = l_i$.
2. Player i then splits $\mathbf{S}_i$ into $(n - t)$-out-of-n Shamir secret shares. Let $\mathbf{X}_{i,j}$ be the j-th share of $\mathbf{S}_i$. Let $\widehat{\mathbf{X}}_{i,j} = \mathsf{comm}(\mathbf{X}_{i,j}, r_{i,j})$ where $r_{i,j}$ are fresh randomness. Broadcast the commitments $\{\widehat{\mathbf{X}}_{i,j}\}_{j\in[n]}$.
3. If a player i fails to broadcast the commitments, add i to the set $\mathcal{D}_s$.

Validation Phase For $sid \in [\lambda]$, let $*^{sid}$ denote the variable $*$ in session sid. Player i invoke $\mathsf{IdealZK}^*[\mathsf{stmt}_i, i, j]$ for each $j \in [n]$, with the statement $\mathsf{stmt}_i = \{\widehat{\mathbf{X}}_{i,j}^{sid}\}_{j\in[n],sid\in[\lambda]}$, and send the witness $w = (m_i, \mathsf{mid}_i, \{\mathbf{S}_i^{sid}\}_{sid\in[\lambda]}, \{\mathbf{X}_{i,j}^{sid}, r_{i,j}^{sid}\}_{j\in[n],sid\in[\lambda]})$ to prove that

- For each $sid \in [\lambda]$, for each $j \in [n]$, $(\mathbf{X}_{i,j}^{sid}, r_{i,j}^{sid})$ is the correct opening of $\widehat{\mathbf{X}}_{i,j}^{sid}$;
- For each $sid \in [\lambda]$, $\{\mathbf{X}_{i,j}^{sid}\}_{j\in[n]}$ forms a valid $(n - t)$-out-of-n secret sharing of $\mathbf{S}_i^{sid}$;
- For each $sid \in [\lambda]$, the vector $\mathbf{S}_i^{sid}$ contains only one non-zero slot $(m_i, \mathsf{mid}_i, 1)$.

For each $i \in [n]$, if there exists a j that $\mathsf{IdealZK}^*[\mathsf{stmt}_i, i, j]$ outputs (fail, i), i.e., the prover fails to prove the statement to receiver j, add i to the set $\mathcal{D}_s^{sid}$ for all $sid \in [\lambda]$.

Sharing phase Continue the following for λ independent, parallel sessions:

1. For $j \in [n]$, player i sends $(\mathbf{X}_{i,j}, r_{i,j})$ to player j.
2. Player i does the following: for every $j \in [n] \setminus \mathcal{D}_s$, if it receives a message $(\mathbf{X}_{j,i}, r_{j,i})$ that is a correct opening with respect to $\widehat{\mathbf{X}}_{j,i}$, record $(\mathbf{X}_{j,i}, r_{j,i})$ and broadcast (ok, i, j). Otherwise, broadcast $(\mathsf{complain}, i, j)$ to complain about j.
3. Player i does the following: for all j such that there is a complain $(\mathsf{complain}, j, i)$ in Step 2, player i broadcasts the corresponding opening $(i, j, \mathbf{X}_{i,j}, r_{i,j})$.

4. Unless player i broadcasts all correct openings for those players who has sent $(\mathsf{complain}, j, i)$ to complain about i, add i to the set $\mathcal{D}_s$.

5. Player i does the following: for $j \in [n] \setminus \mathcal{D}_s$, if player i sent $(\mathsf{complain}, i, j)$ in Step 2, and j broadcast a correct opening $(\mathbf{X}_{j,i}, r_{j,i})$ in Step 3. then record the correct opening $(\mathbf{X}_{j,i}, r_{j,i})$.

Reconstruction Phase Run the following for λ independent, parallel sessions:

1. Player i computes $\mathbf{Y}_i = \sum_{j \in [n] \setminus \mathcal{D}_s} \mathbf{X}_{j,i}$ and broadcast $\mathbf{Y}_i$. If a player j fails to broadcast, add j to the set $\mathcal{D}_r$.

2. Player i does the following for each $j \in [n]$: invoke $\mathsf{IdealZK}^*[\mathsf{stmt}'_i, i, j]$ with the statement $\mathsf{stmt}'_i = (\mathcal{D}_s, \mathbf{Y}_i,\ \{\widehat{\mathbf{X}}_{j,i}\}_{j \in [n] \setminus \mathcal{D}_s})$. It sends the witness $w' = (\{\mathbf{X}_{j,i}, r_{j,i}\}_{j \in [n] \setminus \mathcal{D}_s})$ to the ideal functionality $\mathsf{IdealZK}^*$ to prove that

   - For any $j \in [n] \setminus \mathcal{D}_s$, $(\mathbf{X}_{j,i}, r_{j,i})$ is a correct opening of $\widehat{\mathbf{X}}_{j,i}$;
   - $\mathbf{Y}_i = \sum_{j \in [n] \setminus \mathcal{D}_s} \mathbf{X}_{j,i}$.

3. If there exists a j such that $\mathsf{IdealZK}^*[\mathsf{stmt}'_i, i, j]$ outputs (fail, i), i.e., the prover fails to prove the statement to receiver j, add i to the set $\mathcal{D}_r$.

4. If $|\mathcal{D}_r| \geq t$, everyone stores $(\mathsf{fail}, \mathcal{D}_r \cup \mathcal{D}_s)$ for the reconstruction phase of this session.

5. Otherwise, every player uses all broadcast shares $\{\mathbf{Y}_i\}_{i \in [n] \setminus \mathcal{D}_r}$ to reconstruct the sum $\mathbf{S} = \sum_{i \notin \mathcal{D}_s} \mathbf{Y}_i$. Store $(\mathsf{ok}, \mathbf{S})$ for the reconstruction phase of this session.

Output Phase For each $sid \in [\lambda]$, we use $(\mathsf{fail}, \mathcal{D}^{sid})$ or $(\mathsf{ok}, \mathbf{S}^{sid})$ to denote the value each player stores in the reconstruction phase of session sid. Each player i does the following:

1. If there is a $sid \in [\lambda]$ such that player i stores $(\mathsf{fail}, \mathcal{D}^{sid})$ for that session, outputs $(\mathsf{fail}, \cup_{sid \in [\lambda]} \mathcal{D}^{sid})$, where $\mathcal{D}^{sid} = \emptyset$ for those successfully reconstructed sessions.

2. Otherwise, each player does the following: We say that (m, mid) *appears* in session sid if there exists a slot $l \in [M]$ such that $\mathbf{S}^{sid}[l] = (m, \mathsf{mid}, 1)$.

   For each pair (m, mid) that appears in a majority number of sessions, add a copy of m to Out.

3. Output $(\mathsf{ok}, \mathsf{Out})$.

Theorem 6.2. *If the commitment scheme* comm *is perfectly binding and computationally hiding, then* $\mathsf{Anon}^{t,\mathcal{O}}$ *securely realizes* $\mathcal{F}^{t,\mathcal{O}}_{\text{anon}}$ *in the* $\mathsf{IdealZK}^*$*-hybrid model as long as* $|\mathcal{O}| - t \geq |\mathcal{K}|$. *Moreover,* $\mathsf{Anon}^{t,\mathcal{O}}$ *runs in constant number of rounds.*

Proof. We show that for any non-uniform p.p.t. adversary $\mathcal{A}$ interacting with $\mathsf{Anon}^{t,\mathcal{O}}$, there exists an *admissible* adversary $\mathcal{S}$ interacting with $\mathcal{F}^{t,\mathcal{O}}_{\text{anon}}$, such that $\mathcal{A}$'s views in an execution with $\mathsf{Anon}^{t,\mathcal{O}}$ is computationally indistinguishable from its view simulated by $\mathcal{S}$. Still, we assume that $\mathcal{O} = [n]$. We use $\mathcal{H}$ to denote the set of honest players and $\mathcal{K}$ to denote the set of corrupted players.

The simulator behaves as follows:

Preparation Phase Run the following for λ independent, parallel sessions:

1. Emulate honest player $i \in \mathcal{H}$ as follows: For corrupt player $k \in \mathcal{K}$, let $\mathbf{X}_{i,k}$ be uniformly random chosen. For honest $i' \in \mathcal{H}$, let $\mathbf{X}_{i,i'} = \mathbf{0}$.

2. Emulate honest player $i \in \mathcal{H}$ as follows: commit to the shares $\widehat{\mathbf{X}}_{i,j} = \mathsf{comm}(\mathbf{X}_{i,j}, r_{i,j})$ using fresh randomness $r_{i,j}$ for $j \in [n]$. Send the commitments $\{\widehat{\mathbf{X}}_{i,j}\}_{j\in[n]}$ to $\mathcal{A}$.

3. Wait for corrupted players to send their commitments $\{\widehat{\mathbf{X}}_{k,j}\}_{j\in[n]}$ for $k \in \mathcal{K}$. If a corrupted player k fails to send the commitments, add k to $\mathcal{D}_s$.

Validation Phase Emulate the $\mathsf{IdealZK}^*$ ideal functionality $\mathsf{IdealZK}^*$ as follows:

- For honest prover i and honest verifier i' where $i, i' \in \mathcal{H}$, send $(\mathsf{success}, i, i')$ to $\mathcal{A}$.
- For honest prover $i \in \mathcal{H}$ and corrupt verifier $k \in \mathcal{K}$: If received $\perp$ from a corrupted verifier $k \in \mathcal{K}$, send (fail, k) to $\mathcal{A}$. Otherwise, send $(\mathsf{success}, i, k)$ to $\mathcal{A}$.
- For corrupt prover $k \in \mathcal{K}$ and honest verifier, send ok to $\mathsf{IdealZK}^*$ for the honest verifier, and forward $\perp$ or witness w received from $\mathcal{A}$ to $\mathsf{IdealZK}^*$. Send the output of $\mathsf{IdealZK}^*$ to $\mathcal{A}$.
- For corrupt prover $k \in \mathcal{K}$ and corrupt verifier $k' \in \mathcal{K}$, receive a bit from $\mathcal{A}$, and send the output of $\mathsf{IdealZK}^*$ to $\mathcal{A}$.

For corrupt prover $k \in \mathcal{K}$, if there exists a verifier $j \in [n]$ such that the output of $\mathsf{IdealZK}^*[\mathsf{stmt_k}, k, j]$ is (fail, k), add k to $\mathcal{D}_s^{sid}$ for all $sid \in [\lambda]$. Note that by this point, for $k \in \mathcal{K} \setminus \mathcal{D}_s$, the simulator $\mathcal{S}$ has received the witness m_k, mid_k, $\{\mathbf{S}_k^{sid}\}_{sid\in[\lambda]}$ and $\{\mathbf{X}_{k,j}^{sid}, r_{k,j}^{sid}\}_{j\in[n], sid\in[\lambda]}$.

Sharing Phase Run the following for λ independent, parallel sessions:

1. Emulate the honest players $i \in \mathcal{H}$ to send the shares $\{(\mathbf{X}_{i,k}, r_{i,k})\}_{k\in\mathcal{K}}$ to $\mathcal{A}$.

2. Receive the shares $\{(\mathbf{X}_{k,i}, r_{k,i})\}_{i\in\mathcal{H}}$ from corrupted player $k \in \mathcal{K}$.

3. Emulate honest player i as follows: it checks whether $(\mathbf{X}_{k,i}, r_{k,i})$ it receives from corrupted player $k \in \mathcal{K}$ is a correct opening of $\widehat{\mathbf{X}}_{k,i}$. If yes, send (ok, i, k) to $\mathcal{A}$. Otherwise send $(\mathsf{complain}, i, k)$ to $\mathcal{A}$.

4. Emulate honest player i as follows: If it receives $(\mathsf{complain}, k, i)$ from a corrupt player k, send $(\mathbf{X}_{i,k}, r_{i,k})$ to $\mathcal{A}$.

5. Receive opening $(\mathbf{X}_{k,i}, r_{k,i})$ from corrupt player $k \in \mathcal{K} \setminus \mathcal{D}_s$, if there exists a complain $(\mathsf{complain}, i, k)$ for k.

6. Check for all corrupted players $k \in \mathcal{K} \setminus \mathcal{D}_s$: If k sent all correct openings $(\mathbf{X}_{k,j}, r_{k,j})$ of commitment $\widehat{\mathbf{X}}_{k,j}$ for those j who complained, record the correct openings. Otherwise, add k to the set $\mathcal{D}_s$.

At the end of the sharing phase, for corrupt players $k \in \mathcal{K} \setminus \mathcal{D}_s$, the simulator $\mathcal{S}$ has seen their secrets $\{\mathbf{S}_k^{sid}\}_{sid\in[\lambda]}$ as part of the witness of $\mathsf{IdealZK}^*$. For each $sid \in [\lambda]$, the simulator $\mathcal{S}$ computes $\mathbf{Z}^{sid} = \sum_{k\in\mathcal{K}\setminus\mathcal{D}_s} \mathbf{S}_k^{sid}$. Let Out_K be an initially empty set. $\mathcal{S}$ uniformly randomly picks $|\mathcal{H}|$ slots $\{l_i^{sid}\}_{i\in\mathcal{H}}$ from $[M]$ with replacement. We say that the pair (m, mid) *registers* in session sid if there exists a slot $l \in [M] \setminus \{l_i^{sid}\}_{i\in\mathcal{H}}$, such that $\mathbf{Z}^{sid}[l] = (m, \mathsf{mid}, 1)$. For every $k \in \mathcal{K}$, if (m_k, mid_k) registers in a majority number of sessions, add a copy of m_k to Out_K. Send messages in Out_K and $|\mathcal{K}| - |\mathsf{Out}_K|$ number of $\perp$ to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$.

Reconstruction Phase Receive a set Out of messages from $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Let $\mathsf{Out}_H = \mathsf{Out} \setminus \mathsf{Out}_K$ be the set of honest messages. Then $|\mathsf{Out}_H| = |\mathcal{H}|$. The simulator parses $\mathsf{Out}_H = \{m_i\}_{i\in\mathcal{H}}$. For each honest player i, the simulator uniformly randomly picks a nonce mid_i from $\mathbb{F}$. If there exists an $i \in \mathcal{H}$ and a $j \in [n] \setminus \mathcal{D}_s$ where $j \neq i$, such that $\mathsf{mid}_i = \mathsf{mid}_j$, the simulator aborts.

For each $sid \in [\lambda]$, the simulator computes a vector $\mathbf{S}^{sid}$ as follows: for each message $m_i \in \mathsf{Out}_H$, the simulator adds $(m_i, \mathsf{mid}_i, 1)$ to the slot l_i^{sid} in $\mathbf{Z}^{sid}$. The simulator then checks if for each honest message m_i in Out_H, the pair (m_i, mid_i) appears in a majority number of sessions (Recall that a pair (m, mid) appears in session sid if there exists an $l \in [M]$ such that $\mathbf{S}^{sid}[l] = (m, \mathsf{mid}, 1)$). If not, the simulator aborts.

1. Let $\mathbf{S} = \mathbf{S}^{sid}$ for the current session sid.
2. Compute an $(n-t)$-out-of-n sharing $\{\mathbf{Y}_j\}_{j\in[n]}$ of $\mathbf{S}$ such that for $k \in \mathcal{K}$, $\mathbf{Y}_k = \sum_{j\in[n]\setminus\mathcal{D}_s} \mathbf{X}_{j,k}$. Note that $\mathbf{X}_{i,k}$ for $i \in \mathcal{H}$ are generated by $\mathcal{S}$ in the sharing phase, while $\mathbf{X}_{k',k}$ for $k' \in \mathcal{K} \setminus \mathcal{D}_s$ are received as part of the witness of $\mathsf{IdealZK}^*$ in the sharing phase. This can be done due to the security of Shamir secret sharing and the assumption that $n - t \geq |\mathcal{K}|$.
3. Send $\{\mathbf{Y}_i\}_{i\in\mathcal{H}}$ to $\mathcal{A}$.
4. Wait the corrupt players $k \in \mathcal{K}$ to send $\mathbf{Y}_k$. If a corrupt player k fails to send $\mathbf{Y}_k$, add k to the set $\mathcal{D}_r$.
5. Emulate the $\mathsf{IdealZK}^*$ functionality $\mathsf{IdealZK}^*[\mathsf{stmt}'_i, i, j]$ as follows:
   - For honest prover i and honest verifier i' where $i, i' \in \mathcal{H}$, send $(\mathsf{success}, i, i')$ to $\mathcal{A}$.
   - For honest prover $i \in \mathcal{H}$ and corrupt verifier $k \in \mathcal{K}$: If received $\perp$ from a corrupted verifier $k \in \mathcal{K}$, send (fail, k) to $\mathcal{A}$. Otherwise, send $(\mathsf{success}, i, k)$ to $\mathcal{A}$.
   - For corrupt prover $k \in \mathcal{K}$ and honest verifier, send ok to $\mathsf{IdealZK}^*$ for the honest verifier, and forward $\perp$ or witness w received from $\mathcal{A}$ to $\mathsf{IdealZK}^*$. Send the output of $\mathsf{IdealZK}^*$ to $\mathcal{A}$.
   - For corrupt prover $k \in \mathcal{K}$ and corrupt verifier $k' \in \mathcal{K}$, receive a bit from $\mathcal{A}$, and send the output of $\mathsf{IdealZK}^*$ to $\mathcal{A}$.
6. If $|\mathcal{D}_r| \geq t$, store $(\mathsf{fail}, \mathcal{D}_r \cup \mathcal{D}_s)$ to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ for this session.
7. Otherwise, store ok for this session.

Output phase If there exists a session $sid \in [\lambda]$ that the simulator stores $(\mathsf{fail}, \mathcal{D}^{sid})$, then send $\mathcal{D} = \cup_{sid\in[\lambda]}\mathcal{D}^{sid}$ to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$. Otherwise, send ok to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$.

We first show that the probability that the simulator aborts is negligible.

Claim 6.3. *There exists a negligible function* $\mathsf{negl}(\cdot)$*, such that the probability that the simulator* $\mathcal{S}$ *aborts is at most* $\mathsf{negl}(\lambda)$*.*

Proof. The simulator aborts if either 1) there exists an honest message ID that collides with another message ID; or 2) there exists an honest message m, such that (m, mid_m) does not appear in a majority number of sessions. We now compute the probability of these two events separately.

For the first event, note that the probability that an honest message ID for honest player i collides with another message ID for player j, where $j \in [n] \setminus \mathcal{D}_s$ is $\frac{1}{2^\lambda}$. By a union bound over the total number of pairs of $i \in \mathcal{H}$ and $j \in [n] \setminus \mathcal{D}_s$, the probability that there exists an honest message ID that collides with another message ID is at most $\frac{n(n-1)}{2} \cdot \frac{1}{2^\lambda}$, which is negligible.

For the second event, for any fixed session sid, the probability that an honest message (m, mid_m) does not appear in this session sid is at most $\frac{n}{M} = \frac{1}{2}$. Therefore, the probability that m does not appear in more than $\frac{\lambda}{2}$ sessions is at most $\left(\frac{1}{2}\right)^{\lambda/2}$. By a union bound over the number of honest messages, the probability that there exists an honest message m, such that (m, mid_m) does not appear in a majority number of sessions is at most $n \cdot \left(\frac{1}{2}\right)^{\lambda/2}$, which is negligible.

Therefore, the probability that the simulator aborts is at most $\frac{n(n-1)}{2} \cdot \frac{1}{2^\lambda} + n \cdot \left(\frac{1}{2}\right)^{\lambda/2} = \mathsf{negl}(\lambda)$. □

We now show that the joint distribution of the output of the honest parties and the view of the adversary in the ideal execution (denoted as $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Ideal}}$) is computationally indistinguishable to the output of the honest parties and the view of the adversary in the real execution (denoted as $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anon}}$).

Consider the following hybrids:

1. Hyb_0: denotes an execution of $\mathsf{Anon}^{t,\mathcal{O}}$, in which the simulator acts on behalf of all honest players and interacts with the adversary. Moreover, the simulator emulates $\mathsf{IdealZK}^*$ for the adversary.

2. Hyb_1: The simulator behaves same as in Hyb_1 except that in the preparation phase, for honest player $i \in \mathcal{H}$, it generates $(n-t)$-out-of-n sharing $\{\mathbf{X}_{i,j}\}_{j \in [n]}$ of $\mathbf{S}_i$; but the commitments are computed as follows: $\widehat{\mathbf{X}}_{i,k} = \mathsf{comm}(\mathbf{X}_{i,k}, r_{i,k})$ for $k \in \mathcal{K}$, and $\widehat{\mathbf{X}}_{i,i'} = \mathsf{comm}(\mathbf{0}, r_{i,i'})$ for $i' \in \mathcal{H}$.

   Then, in the validation phase, the simulator emulates the $\mathsf{IdealZK}^*$ functionality and vouches for honest players' commitments as follows:

   - For honest prover i and honest verifier i' where $i, i' \in \mathcal{H}$, send $(\mathsf{success}, i, i')$ to $\mathcal{A}$.
   - For honest prover $i \in \mathcal{H}$ and corrupt verifier $k \in \mathcal{K}$: If received $\perp$ from a corrupted verifier $k \in \mathcal{K}$, send (fail, k) to $\mathcal{A}$. Otherwise, send $(\mathsf{success}, i, k)$ to $\mathcal{A}$.

   During the sharing phase, the simulator acts on behalf of honest player $i \in \mathcal{H}$ and sends the shares $\mathbf{X}_{i,j}$ to player j. However, the honest players do not complain about another honest share. Then in the reconstruction phase, the simulator emulates the $\mathsf{IdealZK}^*$ functionality as in the validation phase.

3. Hyb_2: same as Hyb_1 except that, if there exists an honest message that does not appear in the final output Out, the simulator aborts.

$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anon}} \equiv \mathsf{Hyb}_0$: By definition.

$\mathsf{Hyb}_0 \equiv_c \mathsf{Hyb}_1$: This is due to the computational hiding property of the commitment scheme comm. Consider a sequence of hybrids Hyb_0^i for $i \in \mathcal{H}$. Without loss of generality we just assume that $i \in [|\mathcal{H}|]$.

- Hyb_0^0 is same as Hyb_0.
- Hyb_0^i is same as Hyb_0^{i-1} except that the commitments $\widehat{\mathbf{X}}_{i,j}$ for $j \in \mathcal{H}$ are replaced with commitment of $\mathbf{0}$.
- $\mathsf{Hyb}_0^{|\mathcal{H}|}$ is exactly Hyb_1.

By the computational hiding property of the commitment scheme comm, for any $i \in [|\mathcal{H}|]$, we have $\mathsf{Hyb}_0^i \equiv_c \mathsf{Hyb}_0^{i-1}$. Otherwise, if there exists a non-uniform p.p.t. adversary $\mathcal{D}$ that can distinguish Hyb_0^i from Hyb_0^{i-1} with a non-negligible probability, then we can build a non-uniform p.p.t. adversary $\mathcal{D}'$ that can distinguish the commitments of $\mathbf{0}$ and the commitments of the sharing $\{\mathbf{X}_{i,j}\}_{j\in\mathcal{H}}$ with the same advantage. This breaks the hiding property of comm. Consequently, by hybrids argument, we have that $\mathsf{Hyb}_0 \equiv \mathsf{Hyb}_1$.

$\underline{\mathsf{Hyb}_1 \equiv_c \mathsf{Hyb}_2}$: If the simulator does not abort, then these two experiments are identical. Since the probability that the simulator aborts is negligible, it follows that $\mathsf{Hyb}_1 \equiv_c \mathsf{Hyb}_2$.

$\underline{\mathsf{Hyb}_2 \equiv_c \mathsf{Exp}_{\mathcal{A}}^{\mathsf{Ideal}}}$: We first show that if the simulators in both experiments do not abort, then the views of the adversary in these two experiments are identical. In Hyb_2, the shares $\{\mathbf{X}_{i,j}\}_{j\in[n]}$ are computed based on honest players' inputs $\mathbf{S}_i$; and the reconstruction share $\mathbf{Y}_i$ for honest player i equals $\sum_{j\in[n]\setminus\mathcal{D}_s} \mathbf{X}_{j,i}$. In $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Ideal}}$, however, the shares $\{\mathbf{X}_{i,j}\}_{j\in[n]}$ are independent from honest players' inputs $\mathbf{S}_i$; and the reconstruction share $\mathbf{Y}_i$ is computed based on honest players' messages in Out_H. Since the reconstructed vector $\mathbf{S}$ in both experiments are of the same distribution, and that $n - t \geq |\mathcal{K}|$, by the security of Shamir secret sharing, the views of the adversary in these two experiments are identical, had the simulator not aborted.

Next, we show that given any fixed view of the adversary, the outputs of the honest players in these two experiments are identical if the simulators do not abort in both experiments.

Claim 6.4. *If the simulators do not abort in both experiments, then the honest players' outputs are identical in* Hyb_2 *and* $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Ideal}}$.

Proof. If there is a session $sid \in [\lambda]$ in which the reconstruction fails, i.e., $|\mathcal{D}_r^{sid}| \geq t$, then the honest players in the ideal experiment will output $\mathcal{D}$, the set that $\mathcal{S}$ sends to the ideal functionality $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$ in the output phase, while the honest players in Hyb_1 will also output $\mathcal{D}$, according to the protocol.

If the reconstruction succeeds in every session $sid \in [\lambda]$, the simulator $\mathcal{S}$ sends ok to $\mathcal{F}_{\text{anon}}^{t,\mathcal{O}}$, and the honest players in the ideal experiment output Out. Meanwhile, the honest players in Hyb_2 compute a set Out' based on the reconstructed vectors $\{\mathbf{S}^{sid}\}_{sid\in[\lambda]}$ according to the protocol. We now show that $\mathsf{Out}' = \mathsf{Out}$.

To see that $\mathsf{Out} \subseteq \mathsf{Out}'$, note that every single message $m_i \in \mathsf{Out}_H$ is assigned with a unique message ID mid_i. Since the simulator does not abort, every (m_i, mid_i) appears in a majority number of sessions. This implies that every message m_i in Out_H appears in Out'. In addition, for every message m_k in Out_K, the pair (m_k, mid_k) appears in a majority number of sessions by the construction, and thus m_k belongs to Out'. Henceforth, $\mathsf{Out} \subseteq \mathsf{Out}'$.

On the other hand, every honest message in Out' comes from Out_H. Every corrupt message in Out' must be a subset of Out_K. Otherwise, if there exists a corrupt message m_k in Out' that does not belong to Out_K, then it means that the pair (m_k, mid_k) appears in a majority number of sessions yet it does not register in a majority number of sessions. This can only happen if the simulator adds this exact pair (m_k, mid_k) at the beginning of the reconstruction phase, which is impossible by the guarantee that each honest message ID does not collide with any corrupt message ID. Therefore, $\mathsf{Out}' \subseteq \mathsf{Out}$, and the conclusion follows. □

To conclude, the simulated experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Ideal}}$ is computationally indistinguishable from the real experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anon}}$. □

7 Open Questions

Our work leaves open several natural avenues for future research. We highlight a few of them here:

1. Our leader election protocol does not satisfy the stronger notion called approximate sequential fairness proposed by Chung et al. [CCWS21]. It would be interesting to explore the construction of leader election protocols that satisfy sequential notion of fairness with less than $\log\log n$ round complexity.

2. We do not claim practicality of our construction. We believe it is an exciting future direction to design practical variants of our protocols.

Acknowledgement

This work is in part supported by NSF awards under the grant numbers 2044679 and 1704788, a Packard Fellowship, and a generous gift from Nikolai Mushegian.

References

[Abe99] Masayuki Abe. Mix-networks on permutation networks. In *ASIACRYPT*, 1999.

[ACH11] Gilad Asharov, Ran Canetti, and Carmit Hazay. Towards a game theoretic view of secure computation. In *Eurocrypt*, 2011.

[ADGH06] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *PODC*, 2006.

[ADMM14] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. Secure multiparty computations on bitcoin. In *S&P*, 2014.

[AL11] Gilad Asharov and Yehuda Lindell. Utility dependence in correct and fair rational secret sharing. *Journal of Cryptology*, 24(1):157–202, 2011.

[AO16] Bar Alon and Eran Omri. Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In *TCC*, 2016.

[AS16] Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2016.

[Aum74] Robert J Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, 2001.

[BGKO11] Amos Beimel, Adam Groce, Jonathan Katz, and Ilan Orlov. Fair computation with rational players. *Cryptology ePrint Archive*, 2011.

[BK14] Iddo Bentov and Ranjit Kumaresan. How to use bitcoin to design fair protocols. In *CRYPTO*, 2014.

[BL04] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. *SIAM Journal on Computing*, 33(4):783–818, 2004.

[Blu83] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 1983.

[BOO10] Amos Beimel, Eran Omri, and Ilan Orlov. Protocols for multiparty coin toss with dishonest majority. In *CRYPTO*, 2010.

[CCWS21] Kai-Min Chung, T-H Hubert Chan, Ting Wen, and Elaine Shi. Game-theoretic fairness meets multi-party protocols: The case of leader election. In *CRYPTO*, 2021.

[CGF10] Henry Corrigan-Gibbs and Bryan Ford. Dissent: accountable anonymous group messaging. In *CCS*, 2010.

[CGL+18] Kai-Min Chung, Yue Guo, Wei-Kai Lin, Rafael Pass, and Elaine Shi. Game theoretic notions of fairness in multi-party coin toss. In *TCC*, 2018.

[Cha81] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *STOC*, 1986.

[DHR00] Yevgeniy Dodis, Shai Halevi, and Tal Rabin. A cryptographic solution to a game theoretic problem. In *CRYPTO*, 2000.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, 2004.

[Dod06] Yevgeniy Dodis. Fault-tolerant leader election and collective coin-flipping in the full information model, 2006.

[DR+07] Yevgeniy Dodis, Tal Rabin, et al. Cryptography and game theory. *Algorithmic game theory*, pages 181–207, 2007.

[Fei99] Uriel Feige. Noncryptographic selection protocols. In *FOCS*, 1999.

[GGS] Rati Gelashvili, Guy Goren, and Alexander Spiegelman. Short paper: On game-theoretically-fair leader election.

[GK12] Adam Groce and Jonathan Katz. Fair computation with rational players. In *Eurocrypt*, 2012.

[GKM+13] Juan Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *FOCS*, 2013.

[GKTZ15] Juan Garay, Jonathan Katz, Björn Tackmann, and Vassilis Zikas. How fair is your protocol? a utility-based approach to protocol optimality. In *PODC*, 2015.

[GMW19] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *STOC*. 2019.

[GTZ15] Juan Garay, Björn Tackmann, and Vassilis Zikas. Fair distributed computation of reactive functions. In *DISC*, 2015.

[HT04] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation. In *STOC*, 2004.

[HT14] Iftach Haitner and Eliad Tsfadia. An almost-optimally fair three-party coin-flipping protocol. *STOC*, 2014.

[IML05] Sergei Izmalkov, Silvio Micali, and Matt Lepinski. Rational secure computation and ideal mechanism design. In *FOCS*, 2005.

[Kat08] Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. In *TCC*, 2008.

[KN08] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *TCC*, 2008.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *TCC*, 2008.

[MNS09] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In *TCC*, 2009.

[Nas51] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[OPRV09] Shien Jin Ong, David C Parkes, Alon Rosen, and Salil Vadhan. Fairness with an honest minority and a rational majority. In *TCC*, 2009.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, 2004.

[PS17] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *PODC*, 2017.

[RSZ02] Alexander Russell, Michael Saks, and David Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. *SIAM Journal on Computing*, 31(6):1645–1662, 2002.

[RZ01] Alexander Russell and David Zuckerman. Perfect information leader election in log* n+ o (1) rounds. *Journal of Computer and System Sciences*, 63(4):612–626, 2001.

[SGR99] Paul Syverson, D Goldschlag, and M Reed. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):5, 1999.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[WAS22] Ke Wu, Gilad Asharov, and Elaine Shi. A complete characterization of game-theoretically fair, multi-party coin toss. In *Eurocrypt*, 2022.

[Yao82] Andrew C Yao. Protocols for secure computations. In *FOCS*, 1982.

[ZZZR05] Li Zhuang, Feng Zhou, Ben Y Zhao, and Antony Rowstron. Cashmere: Resilient anonymous routing. In *NSDI*, 2005.

Appendix

A Proof of Theorem 6.1

A.1 Public verifiability

In this section we give the proof of Theorem 6.1. Our $\mathsf{IdealZK}^*$-functionality requires additional public verifiability and identifiable abort on top of the zero-knowledge proof ideal functionality in Pass [Pas04], and we can tailor the proof in Pass [Pas04] to our new $\mathsf{IdealZK}^*$-functionality to have public verifiability and identifiable abort. For completeness we will give the full proof here.

We first introduce the public verifiability definitions we need in the proofs.

A.1.1 Bounded-concurrent public-coin zero-knowledge proof

A public verifiable zero-knowledge proof $(\mathsf{P}, \mathsf{V}, f)$ for a language L consists of a pair of interacting Turing Machines called the prover P and the verifier V, as well as a deterministic, polynomial-time public audit function f. If the prover and the verifier completes the proof successfully and produce a transcript Γ, then $f(x, \Gamma)$ outputs a bit $b \in \{0, 1\}$ as the the acceptance of the proof: $f(x, \Gamma)$ outputs 1 to accept a proof, and 0 to reject a proof. We use $\langle \mathsf{P}(x, z), \mathsf{V}(x) \rangle$ to denote a possibly randomized execution between P with input x, z and V with input x.

It is treated as aborting if a malicious verifier V^* (or prover P^*) sends malformed messages outside the valid range. This guarantees that anyone who observes the transcript can determine whether the verifier (or prover) aborts from the protocol.

Public-coin. An interactive proof $(\mathsf{P}, \mathsf{V}, f)$ is public-coin, if V's messages only contain random strings; Moreover, the acceptance of the proof is computed by $f(x, \Gamma_{\langle \mathsf{P}^*, \mathsf{V} \rangle})$.

Completeness. If $x \in L$, for any non-aborting V^*, $f(x, \Gamma_{\langle \mathsf{P}(x,z), \mathsf{V}^*(x) \rangle}) = 1$.

Soundness. If $x \notin L$, we have that $\Pr[f(x, \Gamma_{\langle \mathsf{P}^*(x,z), \mathsf{V}(x) \rangle}) = 1] \leq \mathsf{negl}(\lambda)$ for any non-aborting prover P^*.

Bounded-concurrent zero-knowledge There exists a strict polynomial-time simulator $\mathcal{S}$, such that for any a-priori known $m = \mathsf{poly}(\lambda)$, for any polynomial time V^*, and every list $\{x_i, w_i\}_{i=1}^m$ where $\mathcal{R}_L(x_i, w_i) = 1$, the view of V^* in an m-times concurrent execution of $(\mathsf{P}, \mathsf{V}^*)$ with input $\{x_i, w_i\}_{i=1}^m$ is computationally indistinguishable from $\mathcal{S}(\mathsf{V}^*, x_1, \ldots, x_m)$.

The zero-knowledge proof introduced by Barak [Bar01] and Pass [Pas04] are bounded-concurrent public verifiable zero-knowledge proofs.

A.1.2 Public verifiable commit-with-extract

A public verifiable commitment-with-extract scheme $(\mathsf{C}, \mathsf{R}, g)$ consists of a pair of interacting Turing machines called the committer C, the receiver R, and a deterministic polynomial-time public audit function g. The commitment protocol contains two phases, the commitment phase and the opening phase. If both phases complete sucessfully and produce some transcript Γ, the audit function $g(\Gamma)$ outputs either a bit $b \in \{0, 1\}$ to accept or $\perp$ to reject. We call b the accepting bit. We use the notation $\langle \mathsf{C}(z), \mathsf{R}(z') \rangle$ to indicate a possibly randomized execution between C with input z and R with input z'.

The valid range of a message in the proof can be deterministically computed in polynomial time, given the previous messages in the interaction. If a malicious committer C^* (or receiver R^*) sends malformed messages outside the valid range, it is treated as aborting. Formally, let $m_1, \dots, m_{2r}$ be the messages that are sent in a commitment scheme (the committer C sends message m_{2i-1} and the receiver R sends message m_{2i}, for $i = 1, \dots, r$). There exist deterministic polynomial-time functions $f_1, \dots, f_{2r}$, each outputs a set $M_i = f_i(m_1, \dots, m_{i-1})$ that serves as the valid range of the i-th message. If $m_i \notin M_i$, then the party who sends m_i is treated as aborting. This guarantees that anyone who observes the transcript can determine whether the committer (or receiver) aborts from the protocol.

Perfect correctness Correctness guarantees that an honest committer always complete the protocol and correctly open its input bit, and will not be stuck by a malicious, non-aborting receiver. Formally, for $b \in \{0,1\}$, for any $\lambda \in \mathbb{N}$, if C is honest and receives input bit b, then $\langle \mathsf{C}(1^\lambda, b), \mathsf{R}^*(1^\lambda) \rangle$ will successfully complete with the accepting bit b with probability 1, for any non-aborting R^*. Moreover, any malicious, non-aborting committer C^* cannot cause an honest receiver R to abort the protocol.

Perfect Binding. Perfect binding guarantees that, if the commit phase is completed, the commitment phase will determine only one bit that can be successfully opened. Formally, let $(\Gamma_c, \Gamma_o) \in \{0,1\}^{\ell(\lambda)}$ be the transcripts of the commitment phase and the opening phase, respectively, where $\ell(\lambda)$ is a fixed polynomial function denoting the maximum length of the transcripts. Then for any $\lambda \in \mathbb{N}$, any transcripts $\Gamma_c, \Gamma_o, \Gamma'_o \in \{0,1\}^{\ell(\lambda)}$, if $g(1^\lambda, \Gamma_c, \Gamma_o) = b$ and $g(1^\lambda, \Gamma_c, \Gamma'_o) = b'$ where $b, b' \in \{0,1\}$, it must be that $b = b'$.

Computationally Hiding. Computationally hiding guarantees that at the end of the commitment phase, the receiver learns only a negligible amount of information about the input that the committer commits to. Formally, let $p(1^\lambda, v)$ denote the probability that R^* outputs 1 at the end of the commitment phase in an execution $\langle \mathsf{C}(1^\lambda, v), \mathsf{R}^*(1^\lambda) \rangle$, then for any non-uniform p.p.t. R^*, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, for every $v_1, v_2 \in \{0,1\}$, $|p(1^\lambda, v_1) - p(1^\lambda, v_2)| \leq \mathsf{negl}(\lambda)$. This should hold even if R^* aborts prior to the end of the commitment phase.

Extractability There exists a strict probabilistic polynomial-time commitment extractor CK, such that for every probabilistic polynomial-time committer C^* and for every input x, auxiliary input y and random tape r, there exists a negligible function $\mathsf{negl}(\cdot)$, such that given input the description of C^*, denoted as $\mathsf{desc}_{|x|}(\mathsf{C}^*)$, and x, y, r, the machine CK outputs a pair, denoted as $(CK_1(\mathsf{desc}_{|x|}(\mathsf{C}^*), x, y, r), CK_2(\mathsf{desc}_{|x|}(\mathsf{C}^*), x, y, r))$, such that

1. $\{CK_1(\mathsf{desc}_{|x|}(\mathsf{C}^*), x, y, r)\}_{x,y,r \in \{0,1\}^*} \equiv_c \{\mathsf{view}_{\mathsf{C}^*}\langle \mathsf{C}^*(x, y, r), \mathsf{R}\rangle\}_{x,y,r \in \{0,1\}^*}$, where the right hand side $\mathsf{view}_{\mathsf{C}^*}\langle \mathsf{C}^*(x, y, r), \mathsf{R}\rangle$ denotes the view of C^* when interacting with an honest receiver R;

2. $\Pr[CK_2(\mathsf{desc}_{|x|}(\mathsf{C}^*), x, y, r) = \mathsf{commit\text{-}value}(CK_1(\mathsf{desc}_{|x|}(\mathsf{C}^*), x, y, r))] > 1 - \mathsf{negl}(\lambda)$, where $\mathsf{commit\text{-}value}(\cdot)$ is a function that takes a commiter's view and output the *unique* value the the view binds to, or $\bot$ if no such value exists.

Although the above definition is defined with respect to bits, it can be generalized to commitments of strings. The commit-with-extract protocol given in Barak and Lindell [BL04] is a public verifiable commit-with-extract protocol. [3]

Now we proceed to prove Theorem 6.1 using a public verifiable and identifiable abort variant of the zero-knowledge proof suggested by Pass [Pas04].

Theorem A.1. *(Restatement of Theorem 6.1). Assume the existence of enhanced trapdoor permutations, and collision-resistant hash functions. Then, given an n-party $\mathsf{IdealZK}^*$-hybrid protocol Π^*, in which the number of total calls to $\mathsf{IdealZK}^*$ is upper bounded by a priori known bound $m = \mathsf{poly}(\lambda)$, some of which can be concurrent, there exists a real-world protocol Π such that the following hold:*

- **Simulatability***: For every real-world non-uniform p.p.t. adversary $\mathcal{A}$ controlling an arbitrary subset of up to $n-1$ players in Π, there exists a non-uniform p.p.t. adversary $\mathcal{A}^*$ in the protocol Π^*, such that for any input $(x_1, \ldots, x_n)$, every auxiliary string $z \in \{0,1\}^*$,*

$$\mathsf{Exec}^{\Pi,\mathcal{A}}(1^\lambda, x_1, \ldots, x_n, z) \equiv_c \mathsf{Exec}^{\Pi^*,\mathcal{A}^*}(1^\lambda, x_1, \ldots, x_n, z).$$

  In the above, the notation $\mathsf{Exec}^{\Pi,\mathcal{A}}$ (or $\mathsf{Exec}^{\Pi^*,\mathcal{A}^*}$) outputs each honest players' outputs as well as the corrupt players' (arbitrary) outputs.

- **Round efficiency***: The round complexity of Π is at most a constant factor worse than that of Π^*.*

To prove Theorem 6.1, we use a similar proof indicated in Pass [Pas04]. We will show that any protocol in the $\mathsf{IdealZK}^*$-hybrid world can be reduced to a protocol in $\mathsf{MemberZK}^*$-hybrid world, which can then be realized under bounded concurrent composition by a real-world protocol.

Roughly speaking, the $\mathsf{MemberZK}^*$ functionality behaves like $\mathsf{IdealZK}^*$, except that it does not require the prover to provide a witness to the statement x. Instead, if it receives (x, b) from the prover, $\mathsf{MemberZK}^*$ sends $(\mathsf{success}, i, j)$ to everyone if $b = 1$ and x is in the language L. Formally, the $\mathsf{MemberZK}^*$ ideal functionality is defined as follows:

Ideal functionality $\mathsf{MemberZK}^*[x, L, i, j]$

The functionality involves n parties $1, \ldots, n$, and is parametrized with the statement x, the language L, the prover's identity i, and the verifier's identity j.

1. If both the prover i and the verifier j are corrupt, receive a bit b from the prover i. If $b = 1$, send $(\mathsf{success}, i, j)$ to everyone.
2. Receive a message ok or $\perp$ from the verifier.
3. If the verifier sends $\perp$, send (fail, j) to everyone.
4. Receive a message $\perp$ or (x, b) from the prover i, where x is the statement and $b \in \{0,1\}$ is a bit.
5. Otherwise if $b = 1$ and $x \in L$, send $(\mathsf{success}, i, j)$ to everyone.
6. Otherwise, send (fail, i) to everyone.

Generally, the $\mathsf{MemberZK}^*$ functionality cannot be computed efficiently, because it is difficult to determine whether $x \in L$ without the witness. Thus, we consider non-abusing adversary, which almost always make proper use of the $\mathsf{MemberZK}^*$ functionality.

[3] The commit-with-extract protocol makes use of a bounded-concurrent public coin zero-knowledge proof. We define that failing a proof is also treated as aborting.

Definition A.2 (Non-abusing adversaries)**.** *We say that an adversary $\mathcal{A}$ in the* $\mathsf{MemberZK}^*$*-hybrid model is non-abusing if, with overwhelming probability, $\mathcal{A}$ only invokes the* $\mathsf{MemberZK}^*$ *functionality with input of (x, b) in Step 4, such that $b = 0$ if $x \notin L$.*

With the restriction to non-abusing adversaries in the $\mathsf{MemberZK}^*$-hybrid model, the $\mathsf{MemberZK}^*$ functionality is efficiently computable.

A.2 Realizing the $\mathsf{MemberZK}^*$-hybrid model

We now show that running the zero-knowledge proof in [Pas04] over the broadcast channel realizes $\mathsf{MemberZK}^*$. Formally, suppose that there exist a collision resistant hash functions ensemble $\{\mathcal{H}_k\}_{h\in\{0,1\}^k}$, as well as a perfect binding, computationally hiding commitment scheme comm. Let $m = \mathsf{poly}(\lambda)$ be an a-priori known bound on the number of concurrent sessions. The special purpose protocols $c\mathcal{ZK}_1, \ldots, c\mathcal{ZK}_m$ each involves n parties, and are parametrized with the statement x, the language L, the prover's identity i and the verifier's identity j. Recall that $c\mathcal{ZK}$ are public-coin, and the acceptance can be computed by a public audit function $f(x, \Gamma)$, where Γ is the transcript of the proof.

Protocol $c\mathcal{ZK}_{sid}[x, L, i, j]$

Input: prover i has input $(1^\lambda, w)$, other players have input 1^λ.

Length parameter: $\ell(\lambda)$.

Set-up:

Verifier j chooses $h \xleftarrow{\$} \mathcal{H}_k$, and posts (h, sid) to the broadcast channel.

Slot 1:

Prover i computes $c_1 = \mathsf{comm}(0^\lambda)$ and posts (c_1, sid) to the broadcast channel.

Verifier j chooses a random challenge $r_1 \xleftarrow{\$} \{0,1\}^{sid\cdot\ell(\lambda)}$, and posts (r_1, sid) to the broadcast channel.

Slot 2:

Prover i computes $c_2 = \mathsf{comm}(0^\lambda)$ and posts (c_2, sid) to the broadcast channel.

Verifier j chooses a random challenge $r_2 \xleftarrow{\$} \{0,1\}^{(m+1-sid)\ell(\lambda)}$, and posts (r_2, sid) to the broadcast channel.

Proof Body

The prover i interact with the verifier j over the broadcast channel to run a witness-indistinguishable universal argument, proving the OR of the following two statements:

1. There exists w such that $\mathcal{R}_L(x, w) = 1$.
2. There exists a triple $\langle \Pi, s, y\rangle$ such that either $\mathcal{R}_{\text{sim}}(\langle h, c_1, r_1\rangle, \langle \Pi, s, y\rangle) = 1$, or $\mathcal{R}_{\text{sim}}(\langle h, c_2, r_2\rangle, \langle \Pi, s, y\rangle) = 1$

Output: Every player auditing the transcript of the proof over the broadcast channel does the following: if the prover aborts from the proof, every player outputs (fail, i); if the verifier aborts from the proof, every player outputs (fail, j). Otherwise if the proof completes and produces transcript Γ, run the public audit function $f(x, \Gamma)$. If $f(x, \Gamma) = 1$, output $(\mathsf{success}, i, j)$. Otherwise if $f(x, \Gamma) = 0$, output (fail, i).

Relation $\mathcal{R}_{\text{sim}}$

Input: A triple $\langle h, c, r \rangle$.

Witness: A program Π, a string $y \in \{0,1\}^{|r|-\lambda}$, a string s.

Relation: $\mathcal{R}_{\text{sim}}(\langle h, c, r \rangle, \langle \Pi, s, y \rangle) = 1$ if and only if

1. $c = \mathsf{comm}(h(\Pi); s)$.
2. $\Pi(c, y) = r$ within $T(\lambda)$ steps.

As mentioned in Section A.1, since this protocol is public-coin, everyone who sees the transcript can determine whether the prover (or verifier) aborts from the protocol, and can compute the acceptance of an honest verifier. Therefore, every player's output is well-defined.

Given the access to the description of the adversary $\mathcal{A}$, the protocol can be simulated. The simulator $\mathcal{S}$ commits to $\mathcal{A}$'s *next-message function*, and prove the $\mathcal{R}_{\text{sim}}$ relation. The next-message function of $\mathcal{A}$ outputs r_1 (or r_2), given the description of all the messages $\mathcal{A}$ received between c_1 and r_1 (or c_2 and r_2). The next-message function, together with the description of these messages that $\mathcal{A}$ receives between c_1 and r_1 (or c_2 and r_2), serve as the witness y of $\mathcal{R}_{\text{sim}}$. Therefore, to prove the $\mathcal{R}_{\text{sim}}$ relation, the simulator needs a *short* description y of length $|r_1| - \lambda$ (or $|r_2| - \lambda$) of all the messages $\mathcal{A}$ receives from the honest players in Slot 1 (or Slot 2). Since the length of all the messages in one execution of $c\mathcal{ZK}$ is bounded by $2\lambda^2$, not counting the *long* random challenge r_1 and r_2, the length parameter is chosen such that $\ell(\lambda) = m \cdot 4\lambda^2 + \mathsf{length}(\Pi') + \lambda$, where m is the upper bound of the total number of concurrent $\mathsf{MemberZK}^*$ calls, and $\mathsf{length}(\Pi')$ is the length of messages of Π' in the $\mathsf{MemberZK}^*$-hybrid model. This guarantees that the simulator have a short description y of all the messages $\mathcal{A}$ receives.

Lemma A.3. *Let Π' be an n-party protocol in the $\mathsf{MemberZK}^*$-hybrid model, in which the number of calls of $\mathsf{MemberZK}^*$ is upper bounded by an a-priori known bound $m = \mathsf{poly}(\lambda)$. Then, assuming the existence of collision resistance hash functions, there exists a protocol Π with the following properties:*

- **Simulatability***: For every real-model non-uniform p.p.t. adversary $\mathcal{A}$ participating in Π, there exists an non-abusing non-uniform p.p.t. adversary $\mathcal{A}'$ interacting with Π', such that for all inputs $(x_1, x_2, \ldots, x_n)$, every auxiliary string $z \in \{0,1\}^*$,*

$$\{\mathsf{Exec}^{\Pi,\mathcal{A}}(1^\lambda, x_1, \ldots, x_n, z)\} \equiv_c \{\mathsf{Exec}^{\Pi',\mathcal{A}'}(1^\lambda, x_1, \ldots, x_n, z)\}.$$

  In the above, $\mathsf{Exec}^{\Pi,\mathcal{A}}$ or ($\mathsf{Exec}^{\Pi',\mathcal{A}'}$) outputs each honest player's outputs as well as the corrupt players' (arbitrary) outputs.

- **Round-efficiency***: The protocol Π uses same round complexity as Π' up to a constant factor.*

Proof. The protocol Π is constructed from protocol Π' by instantiating the m concurrent $\mathsf{MemberZK}^*$ calls with the special-purpose zero-knowledge protocols $c\mathcal{ZK}_1, \ldots, c\mathcal{ZK}_m$.

The simulator $\mathcal{A}'$ acts as follows: given the description of $\mathcal{A}$, its input x and auxiliary input z, $\mathcal{A}'$ internally incorporates $\mathcal{A}$ for Π and externally forward all messages that are not part of the protocols $c\mathcal{ZK}_1, \ldots, c\mathcal{ZK}_m$. Messages for these $c\mathcal{ZK}$ protocols are dealt with internally. At the end of the protocol, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs.

We now proceed to describe how to deal with the messages in $c\mathcal{ZK}$. Each execution of $c\mathcal{ZK}$ is treated independently. We separate the cases when the prover is corrupt and when the prover is honest. Recall that f is the public audit function, that given a complete transcript of the proof, computes the acceptance of the proof.

Both the prover i and the verifier j are corrupted

1. $\mathcal{A}'$ audits the transcript Γ that the prover i and the verifier j broadcast. If both the prover and the verifier do not abort, and that $f(x, \Gamma) = 1$, then $\mathcal{A}'$ sends $b = 1$ to $\mathsf{MemberZK}^*$.
2. Otherwise, if the prover aborts, then $\mathcal{A}'$ sends $\perp$ for the prover to $\mathsf{MemberZK}^*$; if the verifier aborts, then $\mathcal{A}'$ sends $\perp$ for the verifier to $\mathsf{MemberZK}^*$.

Both the prover i and the verifier j are honest

1. Upon receiving $(\mathsf{success}, i, j)$ from $\mathsf{MemberZK}^*$ functionality, $\mathcal{A}'$ emulates an interaction between the honest prover and the honest verifier, by simulating a proof on the code of the honest verifier. In the emulation, instead of sending truly random challenges r_1, r_2 on behalf of the verifier j, it picks a λ-bits string and expands it to the appropriate length using a pseudorandom generator. It uses the expanded string as the challenge.
2. $\mathcal{A}'$ posts the transcript to the broascast channel.

If the prover i is corrupt and the verifier j is honest

1. $\mathcal{A}'$ runs the honest verifier strategy, except that the random challenge r_1, r_2 are generated from pseudorandom generator with a seed length λ.
2. If the prover aborts, $\mathcal{A}'$ sends $\perp$ to $\mathsf{MemberZK}^*$ for the prover i.
3. If the proof completes and the honest verifier strategy accepts the proof, $\mathcal{A}'$ sends $(x, 1)$ to $\mathsf{MemberZK}^*$. Otherwise, $\mathcal{A}'$ sends $(x, 0)$ to $\mathsf{MemberZK}^*$.

If the prover i is honest and the verifier j is corrupt

1. $\mathcal{A}'$ runs the simulator of $c\mathcal{ZK}$ for the verifier j to generate a proof for $\mathcal{A}$.
2. If the verifier aborts the protocol, send $\perp$ to $\mathsf{MemberZK}^*$. Otherwise, send ok to $\mathsf{MemberZK}^*$.

Claim A.4. *The adversary $\mathcal{A}'$ described above is non-abusing. That is, if $x \notin L$, $\mathcal{A}'$ only accepts a proof of x from $\mathcal{A}$ with a negligible probability.*

Proof of Claim A.4. This follows from the simulation soundness of $c\mathcal{ZK}_1, \ldots, c\mathcal{ZK}_m$ proved in Pass [Pas04]. □

The view of $\mathcal{A}$ when interacting in the real-world protocol Π and when interacting with $\mathcal{A}'$ in the $\mathsf{MemberZK}^*$-hybrid model are indistinguishable, following a similar proof as in Pass [Pas04]. □

A.3 Reduce $\mathsf{IdealZK}^*$ to $\mathsf{MemberZK}^*$

In this section we show that a protocol in the $\mathsf{IdealZK}^*$-hybrid model, which contains at most $m = \mathsf{poly}(\lambda)$ number of concurrent $\mathsf{IdealZK}^*$ calls, can be reduced to a protocol in $\mathsf{MemberZK}^*$-hybrid model, with at most m concurrent calls to $\mathsf{MemberZK}^*$.

Lemma A.5. *Let Π^* be an n-party protocol in the $\mathsf{IdealZK}^*$-hybrid model, in which the number of concurrent calls of $\mathsf{IdealZK}^*$ is upper bounded by an a-priori known bound $m = \mathsf{poly}(\lambda)$. Then, assuming the existence of enhanced trapdoor permutation, there exists a protocol Π' in the $\mathsf{MemberZK}^*$-hybrid model, in which the number of concurrent calls of $\mathsf{MemberZK}^*$ is at most m, that satisfies the following properties:*

- **Simulatability***: For every non-abusing non-uniform p.p.t. adversary $\mathcal{A}'$ participating in Π', there exists a non-uniform p.p.t. adversary $\mathcal{A}^*$ interacting with Π^*, such that for all inputs $(x_1, x_2, \ldots, x_n)$, every auxiliary string $z \in \{0,1\}^*$,*
$$\{\mathsf{Exec}^{\Pi',\mathcal{A}'}(1^\lambda, x_1, \ldots, x_n, z)\} \approx \{\mathsf{Exec}^{\Pi^*,\mathcal{A}^*}(1^\lambda, x_1, \ldots, x_n, z)\}.$$
In the above, $\mathsf{Exec}^{\Pi',\mathcal{A}'}$ or ($\mathsf{Exec}^{\Pi^*,\mathcal{A}^*}$) outputs each honest player's outputs as well as the corrupt players' (arbitrary) outputs.

- **Round-efficiency***: The protocol Π' uses same round complexity as Π^* up to a constant factor.*

Proof. The protocol Π' is obtained by replacing the $\mathsf{IdealZK}^*[x, L, i, j]$ in Π^* by the protocol below.

Protocol implementing $\mathsf{IdealZK}^*$ in the $\mathsf{MemberZK}^*$-hybrid model

Input: Player i has input $(1^\lambda, w)$, other players have input 1^λ.

1. The prover i and the verifier j run the commit phase of a public verifiable commit-with-extract protocol $(\mathsf{C}, \mathsf{R}, g)$ in which the prover i commits to the witness w. The interaction of this protocol is over the broadcast channel. Let Γ denote the transcript of the commit phase.

2. Run $\mathsf{MemberZK}^*[\Gamma, L', i, j]$, in which the prover i proves to the verifier j that in Step 1 it has committed to a valid witness w. Formally, $\Gamma \in L'$ iff there exists an opening transcript Γ_o such that $g(\Gamma, \Gamma_o) = w$, and $\mathcal{R}_L(x, w) = 1$.

Output: Every player auditing the transcript computes the following output: If the prover aborts in Step 1, outputs (fail, i); if the verifier aborts in Step 1, outputs (fail, j); Otherwise, output whatever it receives from $\mathsf{MemberZK}^*$.

Since the commit-with-extract is public verifiable, anyone who audits the transcript can determine whether the prover or the verifier aborts. Therefore, the output is well-defined. Let $\mathcal{A}'$ be a $\mathsf{MemberZK}^*$ adversary interacting in protocol Π'. We construct $\mathcal{A}^*$ that internally incorporates $\mathcal{A}'$, given the description of $\mathcal{A}'$, its input x', and the auxiliary input z'. $\mathcal{A}^*$ forward all messages in Π^* that do not belong to the $\mathsf{IdealZK}^*$ call. Messages belong to the $\mathsf{IdealZK}^*$ calls are dealt with internally. At the end of the protocol, $\mathcal{A}^*$ outputs whatever $\mathcal{A}'$ outputs.

We now describe how to deal with the messages belong to the $\mathsf{IdealZK}^*$ call. Let trans be the transcript in Π' right before the $\mathsf{IdealZK}^*$ calls. Let $\mathcal{A}'(x', z', \mathsf{trans})$ denote the residual machine with transcript trans (when it is clear from the context, we abuse the notation and use $\mathcal{A}'$ to denote the residual machine). We separate the cases when the prover is corrupt and when the prover is honest.

Both the prover i and the verifier j are corrupted.

1. $\mathcal{A}^*$ audits the transcript of Step 1. If the prover aborts, send $\perp$ to $\mathsf{IdealZK}^*$ for the prover i; if the verifier aborts, send $\perp$ to $\mathsf{IdealZK}^*$ for the verifier.

2. Otherwise if both the prover and the verifier do not abort, A^* emulates the $\mathsf{MemberZK}^*$ functionality and sends the output of $\mathsf{MemberZK}^*$ to $\mathcal{A}'$.

3. If the output of $\mathsf{MemberZK}^*$ is $(\mathsf{success}, i, j)$, send $b = 1$ to $\mathsf{IdealZK}^*$.

4. Otherwise, if the output is (fail, j), send $\perp$ to $\mathsf{IdealZK}^*$ for the verifier j; if the output is (fail, i), send $\perp$ to $\mathsf{IdealZK}^*$ for the prover i.

Given the construction, it directly follows that the view of $\mathcal{A}'$ when executing the protocol Π' is identical to the view of $\mathcal{A}'$ in the emulation by $\mathcal{A}^*$ in the protocol Π^*. Moreover, the honest players' outputs are identical in these two experiments.

Both the prover i and the verifier j are honest.

1. Upon receiving $(\mathsf{success}, i, j)$ from the $\mathsf{IdealZK}^*$ functionality, emulate the honest prover i and the honest verifier j to run the public verifiable commit-with-extract protocol, in which the prover i commits to 0. Send the transcript to $\mathcal{A}'$.

2. Emulate the $\mathsf{MemberZK}^*$ functionality and send $(\mathsf{success}, i, j)$ to $\mathcal{A}'$.

The view of $\mathcal{A}'$ when executing the protocol Π' is computationally indistinguishable to the view of $\mathcal{A}'$ in the emulation by $\mathcal{A}^*$ in the protocol Π^*, due to the computational hiding property of the commitment. Moreover, the honest players' outputs are identical in these two experiments.

If the prover i is honest and the verifier j is corrupt.

1. $\mathcal{A}^*$ commits to 0 to verifier j using the commit-with-extract protocol. If the verifier aborts, $\mathcal{A}^*$ sends $\perp$ to $\mathsf{IdealZK}^*$ for the verifier j.

2. Otherwise, $\mathcal{A}^*$ emulates the $\mathsf{MemberZK}^*$ functionality. If received $\perp$ from the verifier, send (fail, j) to $\mathcal{A}'$, and send $\perp$ to $\mathsf{IdealZK}^*$ for the verifier j.

3. If received ok from the verifier, $\mathcal{A}^*$ emulates the $\mathsf{MemberZK}^*$ functionality and send $(\mathsf{success}, i, j)$ to $\mathcal{A}'$. It then sends ok to $\mathsf{IdealZK}^*$.

The view of $\mathcal{A}'$ when executing the protocol Π' is computationally indistinguishable to the view of $\mathcal{A}'$ in the emulation by $\mathcal{A}^*$ in the protocol Π^*, due to the computationally hiding property of the commit-with-extract commitment. Moreover, the honest players' outputs are also computationally indistinguishable in these two experiments.

If the prover i is corrupt and the verifier j is honest.

1. $\mathcal{A}^*$ interacts in the commit-with-extract phase as the receiver with the residual machine $\mathcal{A}'(x', z', \mathsf{trans})$. If the prover i aborts, $\mathcal{A}^*$ sends $\perp$ to $\mathsf{IdealZK}^*$ for the prover i. Let Γ_c denote the transcript for the commit phase.

2. $\mathcal{A}^*$ then emulates the $\mathsf{MemberZK}^*$ functionality and sends the output to $\mathcal{A}'$.

3. If $\mathsf{MemberZK}^*$ outputs $(\mathsf{success}, i, j)$, then $\mathcal{A}^*$ runs the extractor CK of the commit-with-extract scheme on input $(\mathsf{desc}_{|x'|}(\mathcal{A}'), x', z', \mathsf{trans})$ to obtain a view $\widetilde{v}$, along with a string $\widetilde{w}$ that is the unique value that the view $\widetilde{v}$ binds to. It then sends $\widetilde{w}$ to $\mathsf{IdealZK}^*$ and reset the view of $\mathcal{A}'$ of commitment phase to $\widetilde{v}$.

4. Otherwise if $\mathsf{MemberZK}^*$ outputs (fail, i), sends $\perp$ to $\mathsf{IdealZK}^*$ for the prover i.

The view of $\mathcal{A}'$ when executing the protocol Π' is identical to the view of $\mathcal{A}'$ in the emulation by $\mathcal{A}^*$ in the protocol Π^*. Now we show that the honest players' outputs in these two experiments are computationally indistinguishable. To show this, it suffices to prove that when the $\mathsf{MemberZK}^*$ functionality outputs $(\mathsf{success}, i, j)$, the extracted $\widetilde{w}$ is indeed a valid witness of x with overwhelming probability.

Since $\mathcal{A}$ is non-abusing, the output $(\mathsf{success}, i, j)$ implies that the prover i sends $(\Gamma_w, 1)$ to $\mathsf{MemberZK}^*$, where Γ_w is the transcript of Step 1, in which the residual machine $\mathcal{A}'$ commits to w. Hence, the probability that the prover i commits to a valid witness in the commit phase is $1 - \mathsf{negl}(\lambda)$ in the real execution.

Consider a modified simulator $\widetilde{\mathcal{A}}^*$ that runs the extractor CK and obtains a pair $(\widetilde{v}, \widetilde{w})$. It then obtains the input to $\mathsf{MemberZK}^*$ of prover i defined by the view $\widetilde{v}$. Then, $\widetilde{\mathcal{A}}^*$ sends $\widetilde{w}$ to $\mathsf{IdealZK}^*$ if and only if the emulated input to $\mathsf{MemberZK}^*$ is $(\Gamma_{\widetilde{w}}, 1)$. Since $\widetilde{\mathcal{A}}^*$ contains only additional checks, $\mathcal{A}^*$ sends $\widetilde{w}$ to $\mathsf{IdealZK}^*$ whenever $\widetilde{\mathcal{A}}^*$ sends $\widetilde{w}$ to $\mathsf{IdealZK}^*$. Now it suffices to show that $\widetilde{\mathcal{A}}^*$ sends $\widetilde{w}$ to $\mathsf{IdealZK}^*$ with an overwhelming probability.

By the extractability, $\widetilde{v}$ is indistinguishable from the view of prover i in the execution emulated by $\mathcal{A}^*$, the probability that the emulated input to $\mathsf{MemberZK}^*$ (computed by $\widetilde{\mathcal{A}}^*$) is $(\Gamma_{\widetilde{w}}, 1)$ must be negligibly close to the probability that prover i sends $(\Gamma_w, 1)$ to $\mathsf{MemberZK}^*$. (Otherwise, it is possible to distinguish between the real view of $\mathcal{A}'$ and the view output by CK by emulating the input of prover i to $\mathsf{MemberZK}^*$. This emulation can be carried out because prover i is polynomial-time, and the description of prover i as well as the view $\widetilde{v}$ fully define the residual prover). Again, since prover i is non-abusing, if the emulated input to $\mathsf{MemberZK}^*$ is $(\Gamma_{\widetilde{w}}, 1)$, then $\widetilde{w}$ must be a valid witness with $1 - \mathsf{negl}(\lambda)$ probability. Therefore, given that the prover i sends $(\Gamma_w, 1)$ to $\mathsf{MemberZK}^*$ in the execution simulated by $\mathcal{A}^*$, the probability that the extracted $\widetilde{w}$ is a valid witness is at least $1 - \mathsf{negl}(\lambda)$.

To conclude, the view of $\mathcal{A}'$ when interacting with Π' in the $\mathsf{MemberZK}^*$-hybrid model is indistinguishable from its view simulated by $\mathcal{A}^*$ in the $\mathsf{IdealZK}^*$-hybrid model. □