Game-Theoretically Fair Distributed Sampling [*]

Sri AravindaKrishnan Thyagarajan[†], Ke Wu[‡], Pratik Soni[§]

Abstract

Cleve's celebrated result (STOC'86) showed that a strongly fair multi-party coin-toss is impossible against majority-sized coalitions. Recently, however, a fascinating line of work studied a relaxed fairness notion called *game-theoretic fairness*, which guarantees that no coalition should be incentivized to deviate from the prescribed protocol. A sequence of works has explored the feasibility of game-theoretic fairness for *two-sided* coin-toss, and demonstrated feasibility in the dishonest majority setting under standard cryptographic assumptions. However, this line of work only focused on uniform two-sided coin-toss or leader election.

In this work, we *initiate* the comprehensive study of game-theoretic fairness for multi-party *sampling from general distributions*. In particular, for the case of m-sided *uniform* coin-toss we give a nearly complete characterization of the regime in which game-theoretic fairness is feasible. Interestingly, contrary to standard fairness notions in cryptography, the composition of game-theoretically fair two-sided coin-toss protocols does not necessarily yield game-theoretically fair multi-sided coins. To circumvent this, we introduce new techniques compatible with game-theoretic fairness. In particular, we give the following results:

- We give a protocol from standard cryptographic assumptions that achieves game-theoretic fairness for uniform m-sided coin-toss against half- or more-sized adversarial coalitions.
- To complement our protocol, we give a general impossibility result that establishes the optimality of our protocol for a broad range of parameters modulo an additive constant. Even in the worst-case, the gap between our protocol and our impossibility result is only a small constant multiplicative factor.
- We also present a game-theoretically fair protocol for *any* efficiently sampleable m-outcome distribution in the dishonest majority setting. For instance, even for the case of $m = 2$ (i.e., two-sided coin-toss), our result implies a game-theoretically fair protocol for an *arbitrary* Bernoulli coin. In contrast, the work of Wu, Asharov, and Shi only focussed on a Bernoulli coin with parameter 1/2.

[*] The author order is randomized

[†] University of Sydney, t.srikrishnan@gmail.com

[‡] Carnegie Mellon University kew2@andrew.cmu.edu

[§] University of Utah, psoni@cs.utah.edu

Contents

1 Introduction

Given a distribution, can a group of *mutually distrusting* participants collaborate over the internet to sample from it? This question has been rigorously studied in several avatars across Cryptography and Distributed Computing. Two prominent examples are that of (a) coin-toss, which has a rich history in Cryptography [AO16, BOO15, BHMO22, BHT18, Cle86, HT14, HO14, MNS16, BHLT17, DSLMM11], and (b) leader election which has profound applications in developing consensus protocols [RZ01, RSZ02, Fei99, Dod06, BK14a, ADMM14, ADGH06a]. But what security can be guaranteed?

Let's consider the case of coin-toss. Here, n parties collaboratively want to toss a uniform coin or equivalently sample from the uniform distribution over the set $\{0, 1\}$ (denoted by $U_{\{0,1\}}$). Ideally, for some corruption parameter $t < n$, we would like to build an n-party protocol for coin-toss whose output, as a distribution, is indistinguishable from $U_{\{0,1\}}$, even when any subset of t parties choose to deviate arbitrarily from the protocol (or in other words, behave maliciously). This requirement is formalized in the cryptography literature as *strong fairness.* Clearly, a strongly-fair protocol would be immediately helpful in numerous settings. Unfortunately, the story of building such protocols heavily depends on the corruption parameter t. In the honest majority setting, i.e., $t < n/2$, strongly fair coin-toss protocols are known under standard cryptographic assumptions [Yao82, MGW87, GMW19]. On the contrary, for the dishonest majority setting, i.e., $t \geq n/2$, Cleve [Cle86] showed a strong impossibility result and in particular, rules out the possibility of a strongly-fair protocol for n-party coin-toss in the dishonest majority setting.

To get a glimpse into the challenge of the dishonest majority setting, let's consider Blum's celebrated coin-toss protocol for two parties [Blu83]: parties P_0 and P_1 respectively commit to random bits b_0 and b_1 using cryptographic commitment (i.e., digital analogs of physical locked boxes), and later open their commitments to compute the output $b = b_0 \oplus b_1$. If one of the parties aborts or does not open their commitment, the protocol ends without a coin. For this protocol, there exists a malicious P_1^* that can bias the output distribution by selectively aborting before opening its commitment (that is, P_1 chooses not to open its commitment if it does not "like" the outcome $b_0 \oplus b_1$). In light of this, one might wonder if there is any hope of achieving a weaker yet meaningful notion of fairness for the dishonest majority setting.

Game-theoretic Fairness. While malicious security is the gold standard, in several settings, a real-world adversary may not be incentivized to deviate arbitrarily without hurting its own *interests.* However, it was not clear how to leverage this seemingly intuitive insight until the very recent work of Chung, Guo, Lin, Pass, and Shi (henceforth, CGLPS18) [CGL$^+$18a]. They observed that in the setting where P_0 and P_1 want to use the coin-toss to elect a winner among them, there is implicit a game-theoretic notion of *utility* for each party. More formally, for party P_b, given their public preference towards an outcome (i.e., b), we can define the following binary utility function $u_b(c)$ that is 1 iff the coin output c equals b.

In this setting, a party is disincentivized to adopt *any* malicious strategies that decrease its utility. To leverage this, CGLPS18 cleverly modified Blum's protocol to ensure *any* malicious strategy (including selective aborts) would necessarily result in decreased utility. In particular, their modified protocol is identical to Blum's except that if party P_b aborts, then the output of the coin is set to $1 - b$.

Note that P_1^* adopting the selective abort strategy from above will end up biasing the output in P_0's favor. More generally, (the modified) Blum's protocol ensures that neither of the parties (i.e., a dishonest majority) can increase their expected utility over $\frac{1}{2}$, which is their expected utility when behaving honestly.

Game-theoretically Fair Multi-Party Coin-toss. The feasibility of this game-theoretic notion of fairness for Blum's two-party protocol kicked off a line of research on coin-toss in the *multi-party* setting. Here, CGLPS18 put forth several notions of game-theoretic fairness for the multi-party setting, of which the most relevant notion to our work is that of *Cooperative-Strategy-Proofness* or CSP-fairness for short. Here, each of the n parties has a *publicly announced preference* towards 0 or 1, and it gains a utility of 1 if and only if the coin is consistent with their preference. Similarly, the joint utility of a coalition of parties is the sum of their utilities. Then, CSP-fairness ensures that (a) when all parties are honest, the resulting output is a uniform coin, and (b), more importantly, no adversarial coalition of t parties can increase its expected utility, irrespective of the adopted strategy. Intuitively, CSP-fairness guarantees that any profit-seeking coalition does not want to deviate from the honest protocol. This is a natural notion to consider in real-world applications [CMST22] where players have public preferences.

CGLPS18 gave a simple protocol that satisfies CSP-fairness against $(n-1)$-sized coalition only when exactly one user prefers 0 and all other players prefer 1. In a follow-up work, Wu, Asharov, and Shi [WAS22] give a complete characterization of the feasibility of CSP-fairness for multi-party coin-toss for all preference profiles: Let n_b be the number of parties preferring the outcome b. For the case of "balanced" profiles (i.e., $n_0 = n_1$), they showed an $O(n)$-round protocol that satisfies CSP fairness for any $t = \lceil n/2 \rceil$ under standard cryptographic assumptions, and proved an impossibility result for $t > \lceil n/2 \rceil$ to demonstrate optimality. For the case of "unbalanced" profiles (i.e., n_1 is much larger than n_0), a significantly larger t can be tolerated. However, the exact value of optimal t depends on the exact relation between n_0, n_1.

Our Work: CSP-fair Multi-Party Sampling. While the case of coin-toss with multiple parties is well understood, the broader question of sampling from general distributions is still unexplored. Consider the case of n parties wanting to collaboratively "toss" an m-sided coin as a motivating example. The work of CGLPS18 explores a special case of $m = n$ with a very special preference profile: each of the n parties prefers itself (in other words, a leader election). For this exceptional setting, they give a $O(\log n)$-round CSP-fair protocol when $t = n - 1$, which is optimal. Later works on leader election improve the round complexity at the cost of relaxing the notion to an "approximate" fairness [CCWS21, KMSW22]. However, even for n-sided coin-toss, the case of more general preference profiles has not yet been considered.

This work addresses the feasibility of CSP-fair m-sided coin-toss among n parties. We consider the general task of distributed sampling – where n parties want to sample from a given efficiently sampleable distribution $\mathcal{D}$ (e.g., arbitrary m-sided coin), and focus on the following question:

Can we build multi-party sampling protocols for distribution $\mathcal{D}$ that satisfy CSP-fairness against dishonest majority coalitions?

Following Wu, Asharov, and Shi [WAS22], we aim to completely characterize the feasibility spectrum for a broad class of preference profiles. We emphasize that even for the restricted case of sampling a uniform m-sided coin, the space of preference profiles in our setting is exponentially larger than those of uniform 2-sided coin-toss and the works of leader election. This makes obtaining a characterization like that of [WAS22] significantly more challenging.

1.1 Our Contributions

To describe our contributions, we recall our setting: Let $\mathcal{D}$ be an efficiently sampleable distribution, n be the number of parties $P_1, \ldots, P_n$, and t be the size of the coalition. We conceive of a setting where each of the n parties has a publicly announced preference towards exactly one outcome in the support of $\mathcal{D}$. In terms of utility, we adopt the formalism from the case of binary coin-tossing.

In particular, we assign the party a utility of 1 if the outcome matches its preference, and of 0 otherwise. Similarly, the utility of a coalition is defined as the sum of the utilites of its members. In this setting, our goal is to develop game-theoretically fair sampling protocols in the *plain model* where the parties do not have any form of setup (e.g., a trusted reference string).

The setting of public preferences and each user preferring exactly one outcome might appear too simplistic and restrictive at a first glance. We emphasize that simple setting appears naturally in several applications and potentially serve as fundamental building block for the general case where parties have (equally) prefer several outcomes. We elaborate on this in Section 1.2.

Feasibility for Multi-sided Coin-Toss. First, we discuss the case of general m-sided coin-toss where the n parties want to sample a *uniform* value from the range $\{0, \ldots, m-1\}$. We first consider the case when $m > n$. Since each party prefers one outcome, there exists an outcome that no party prefers, say i^*. In this setting, we give a CSP-fair protocol that is optimal in t, that is, tolerates $t = n - 1$. At a high level, we design a protocol that can detect all malicious behaviors, in which case the protocol's output is set to i^* as punishment. Our protocol has $O(1)$ rounds. It relies on the existence of enhanced trapdoor functions and collision-resistant hash functions, which are standard cryptographic tools and can be built from a variety of well-studied number-theoretic assumptions. In fact, we can extend the protocol to any m for *degenerate preference profiles*: there exists an outcome from $\{0, \ldots, m - 1\}$ that none of the n parties prefer.

Theorem 1.1 (Feasibility for Degenerate Preference Profile)**.** *Assuming the existence of enhanced trapdoor functions and collision-resistant hash functions, for any m and any degenerate preference profile for n parties, there is an $O(1)$-round CSP-fair protocol for m-sided coin-toss tolerating $t = n - 1$ corruptions.*

Next, we consider the case of "non-degenerate" preference profiles where each outcome has at least one supporter. Without loss of generality, let's assume that 0 is the least preferred outcome. For such profiles, we give an $O(n)$-round CSP-fair protocol tolerating dishonest majority coalitions relying on the existence of oblivious transfer. The exact corruption parameter t we tolerate depends on the exact relation between the number of parties preferring the outcome 0, denoted by n_0, the total number of parties n, and the number of sides m. For example, when n and n_0 are reasonably close (i.e., $n = m \cdot n_0$), we can tolerate up to $\lceil n/2 \rceil$ corruptions but can tolerate up to even $0.9 \cdot n$ corruptions for n much larger than $m \cdot n_0$ (say 10 times larger). We formalize the exact achieved parameters in the theorem below:

Theorem 1.2 (Feasibility for Non-Degenerate Preference Profile)**.** *Assuming the existence of oblivious transfer, for any m and any non-degenerate preference profile, there exists a $O(n)$-round CSP-fair protocol for m-sided coin-toss tolerating t corruptions, where*

$$t = \begin{cases} n - \left\lfloor \frac{(2m-1)\cdot n_0}{2} \right\rfloor, & \textit{if } n \geq \frac{(4m-1)\cdot n_0}{2}, \\ \left\lfloor \frac{m}{2m-1} n - \frac{m}{2(2m-1)} n_0 \right\rfloor + (n_0 \mod 2), & \textit{otherwise} . \end{cases} \tag{1}$$

Impossibility for Multi-sided Coin-Toss. To complement our protocols, we also give an impossibility result for a natural class of preference profiles. Without loss of generality, let's assume that $n_0 \leq n_1 \leq \ldots \leq n_{m-1}$ where n_i is the number of parties preferring the outcome i. At a high level, our result follows a similar strategy to that of Wu, Asharov, and Shi's impossibility result for two-sided coin, which essentially reduces to Cleve's impossibility result for two-sided coin-toss. However, to reduce to the two-sided case, we cluster all the m outcomes into two partitions: outcomes smaller than b into one partition, and outcomes larger than or equal to b for some

$b \in \{1, \ldots, m-1\}$ into the other. Here, b is chosen such that parties preferring any outcome $c < b$ are similar in number, and parties preferring any outcome $c \geq b$ are similar in number. We refer to such preference profiles as b-*bipartite* preference profile, which we formally define in Definition 8.2.

Theorem 1.3 (Impossibility). *Let n_i be the number of parties preferring the outcome i. Given any b-bipartite preference profiles $[n_0, \ldots, n_{m-1}]$, there is no efficient protocol for m-sided coin-toss that achieves CSP-fairness against any t-sized coalition, where t equals*

$$b \cdot \max\left\{n_{b-1} - n_0 + 1, \left\lceil \frac{n_{b-1}}{2} \right\rceil\right\} + (m-b) \cdot \max\left\{n_{m-1} - n_0 + 1, \left\lceil \frac{n_{m-1}}{2} \right\rceil + 1\right\}.$$

Unlike the work of Wu, Asharov, and Shi, our impossibility result, in the worst case, is tight only up to a constant multiplicative factor. However, for a broad range of preference profiles, the tightness is off only by a constant additive factor, and in some cases, it is optimal. Therefore, our results can be viewed as giving a *nearly* complete characterization of CSP-fairness for general coin-toss.

Some concrete parameters. We discuss some concrete examples here to better illustrate the gap between our feasibility and impossibility results.

- When $n = m \cdot n_0$, i.e., all n_i's are equal: This profile is $(m-1)$-bipartite. For such profiles, our protocol tolerates any $\lceil \frac{n}{2} \rceil$-sized coalition, and no protocol tolerates coalitions of size $m \cdot \lceil \frac{n_0}{2} \rceil + 1$. Therefore, our protocol is optimal for even n_0 and almost optimal for odd n_0 (except for an additive gap of m).
- When n is much larger than $m \cdot n_0$ (say 10 times larger): our protocol tolerates $0.9n$-sized coalitions, and Theorem 1.3 implies that for bi-partite preference profiles, our protocol is almost optimal except for an additive gap of $(m-1)(n_{m-1} - n_1 + 1)$, which is roughly some constant fraction of n.
- When $n_0 = 1$, our protocol tolerates $(n-m)$-sized coalition, which is nearly optimal for constant m.

On Feasibility for General Distributions. We also extend our feasibility result to the case of general m-outcome distributions. An alternative formulation here is to toss a "non-uniform" m-sided coin where the probability p_i of each side i may not be the same (but sum up to 1). We denote such m-outcome distributions as $\{p_0, \ldots, p_{m-1}\}$, where p_i is the probability of getting the i-th outcome. To describe our feasibility result, we must introduce a new parameter p_j^* for each outcome j. Roughly speaking, this parameter p_j^* measures the maximum fraction of j-supporters in a coalition, such that biasing the output towards j does not increase the coalition's utility [1]. Without loss of generality, assume that outcome 0 maximizes the quantity $p_j^* \cdot n - n_j$.

Theorem 1.4. *Given any m-outcome distribution $\mathcal{D}$,*

- *Assuming the existence of oblivious transfer, for any non-degenerate preference profile, there exists an $O(n)$-round CSP-fair protocol for sampling from $\mathcal{D}$ tolerating t corruptions, where $p^* = p_0^*$, and*

$$t = \begin{cases} n - \left\lfloor \frac{2-p^*}{2p^*} n_0 \right\rfloor + (n_0 \mod 2), & \text{if } n \geq \frac{4-p^*}{2p^*} n_0 \\ \left\lfloor \frac{1}{2-p^*} n - \frac{1}{2(2-p^*)} n_0 \right\rfloor + (n_0 \mod 2), & \text{otherwise,} \end{cases} \tag{2}$$

[1] We refer to the parameter p_j^* as the *balancing* parameter in the technical sections and is slightly different and more involved than the description here.

- *Assuming the existence of enhanced trapdoor permutations and collision-resistant hash functions, for any degenerate preference profile, there exists a $O(1)$-round CSP-fair protocol for sampling from $\mathcal{D}$ tolerating $(n-1)$-sized coalitions.*

For example, for any 2-outcome distribution, our protocol tolerates the same-sized coalition as that for uniform binary coin-toss. For any m-outcome distribution where $p^* = \frac{1}{m-1}$, our protocol tolerates the same-sized coalition as that for uniform m-sided coin-toss. For the case where $p^* < \frac{1}{m-1}$, our protocol tolerates smaller-sized coalition compared to the uniform m-sided coin-toss. However, as long as $p^* \cdot n \geq n_0$, our protocol tolerates half- or more-sized coalition.

1.2 Applicability of Our Model

Recall that our protocols offer a natural notion of game-theoretic fairness in a setting where (a) each party prefers exactly one outcome, henceforth referred to as *single-outcome preference*, and (b) every party's preference is publicly known. We find these requirements natural and in line with our goal to *initiate* the study of game-theoretically fair protocols for the general task of sampling. We elaborate on these aspects below. We emphasize that all prior work in related models relied on these restrictions for the special case of binary coin-tossing and leader election protocols. On the contrary, we discuss how our general sampling protocols with single-outcome preferences have interesting implications for an important generalization of leader election – which has not been explored prior to our work.

Towards General Leader Election. Electing a leader uniformly at random from a set of n parties serves as an important building block in designing consensus algorithms. Existing research on game-theoretically fair leader election protocols only focused on the model where each party exclusively favors itself. However, in current blockchain settings, parties may belong to some mining pools, and they can derive utility when any pool member is elected. In such frameworks, parties have multiple-outcome preferences. Unfortunately, current protocols fail to handle this setting. While our model for single-outcome preference does not directly handle the above scenarios, our new protocol for m-sided coin-toss can serve as a building block for the above general leader election.

To illustrate, consider a natural model with n parties and m mining pools, each party affiliated with a single mining pool. Each party gets utility 1 if the selected leader belongs to its mining pool. The system designer aims to devise a protocol that uniformly selects a leader from the pool of n leaders for the purpose of equity. One approach is to first run our n-party m-sided *non-uniform* coin-tossing protocol to select one mining pool, where the probability is determined by the weight of each pool, followed by a standard single-preference leader election protocol within the selected mining pool. Alternatively, a system designer, motivated to discourage the formation of large mining pools, may opt for an m-sided uniform coin toss (followed by leader election among members of the chosen pool) to select a leader. If the leader election is done this way, then all mining pools have the same chance of getting selected. So, a party is better off creating its own mining pool and being its sole member.

On Public Preferences. Public preference profiles appear naturally in decentralized systems such as blockchains, where parties transparently reveal their preferences when voting on updates or governance decisions. This transparency is often facilitated through public forums, discussions, or stakes in decisions known to the public. For example, voting against a fork in blockchain scenarios can impact a party's stake or reputation. Additionally, in collective games with a random coin, players are aware of each other's preferences, influencing their likelihood of winning. Within these contexts, parties typically prefer a specific outcome that offers a competitive advantage. It is worth noting that designing game-theoretically fair protocols for private preference profiles is an intriguing direction for future research. We believe that addressing this challenge necessitates new techniques compared to recent works, including ours, where the protocol crucially depends on the knowledge of these preference profiles.

On the Coalition Utility. Our model adopts a natural coalition utility as the sum of individual utilities. This notion, also used in previous works, captures the wide range of applications where utility signifies divisible goods, such as currency, wealth, or physical items. In these frameworks, a coalition wants to increase its joint utility compared to honest behaviors, such that there exists a distribution of wealth that gives every coalition member strictly higher utility. We believe our techniques can be extended to other natural ways of defining utility notions for a coalition. We leave an in-depth investigation of this for future work.

1.3 Additional Related Work

There was a line of work [HT04, KN08, ADGH06b, OPRV09, AL11, ACH11] exploring the connection between game theory and multi-party computation (See the survey by Katz [Kat08] and by Dodis and Rabin [DR07])). They adopted a different notion of utility compared to CSP fairness. Specifically, they define the utility functions based on the assumption that players prefer to compute the function correctly and learn others' secrets while not leaking their own secrets. Following this line of work, a new paradigm called the Rational Protocol Design was proposed and further developed in [GKM$^+$13, GKTZ15, GTZ15].

Recently, with the success of blockchains and decentralized applications, many works explored a financial fairness notion [BK14b, KMS$^+$16, ADMM16, KB14, KVV16, DEF18, ATM$^+$22, CMST22]. These works incentivize honesty by using collaterals: players are required to put in collaterals and get punished if they misbehave. Interestingly, our work and the previous work on game-theoretic fairness [CGL$^+$18b, WAS22, CCWS21, KMSW22] guarantees incentive compatibility even without the use of any collateral.

Organization. In Section 2, we give a roadmap to highlight our technical contributions. The preliminaries and the model is given in Section 3. Section 4 and Section 5 present the protocols against semi-malicious coalitions for uniform m-sided coin-toss and general m-outcome distributions. The protocol for degenerate preference profiles is given in Section 6. In Section 7, we show how to upgrade the security of our protocol to achieve CSP-fairness against malicious coalitions. Finally, We give the impossibility result in Section 8.

2 Technical Roadmap

This technical roadmap focuses on characterizing CSP-fairness against semi-malicious coalitions. Intuitively, a semi-malicious adversary follows the protocol description honestly except that 1.) arbitrarily chooses its own randomness; and 2.) may abort the protocol anytime. We can generically upgrade our protocols against malicious coalitions using standard cryptographic techniques.

This section is organized as follows: First, in Section 2.1, we illustrate the failure of composition in the context of CSP-fairness through the example of uniform 4-sided coin-toss. To introduce our technical ideas, we will focus on the case of ternary coin-toss, but all our results generalize to the m-sided coin-toss. In particular, we discuss uniform ternary coin-toss in Section 2.2, in Section 2.3, we focus on the case of ternary coin-toss for general distributions, and give intuitions behind our impossibility result in Section 2.4.

Model and notations. We briefly discuss our model and notations to understand the roadmap better. A more detailed description can be found in Section 3. We assume that players can communicate with each other through a pairwise private channel as well as a public broadcast channel. At the end of the protocol, the output of the protocol is determined by all broadcast messages.

We refer to the players preferring outcome j as j-supporters. Given a desired ternary-coin distribution $\mathcal{D} = \{p_0, p_1, p_2\}$, where $p_j := \Pr[\text{ourcome} = j]$, and a preference profile $[n_0, n_1, n_2]$ where n_j is the number of j-supporters. The coin-toss protocol should satisfy the following properties: 1.) *δ-Correctness:* When everyone behaves honestly, the probability of getting output j is 2^{δ} close to p_j for any $j \in \{0, 1, 2\}$; 2.) *CSP-fairness:* Any t-sized coalition cannot increase its expected utility by deviating.

2.1 Failure of Composition

A natural approach to construct a CSP-fair m-sided coin-toss is to appropriately compose several executions of a CSP-fair binary coin-toss protocol. Imagine that there exists such a binary coin-toss protocol $\Pi(\widetilde{n}_0, \widetilde{n}_1)$ among $\widetilde{n}_0$ number of 0-supporters and $\widetilde{n}_1$ number of 1-supporters. To disambiguate the m-sided outcomes from the binary outcomes, we henceforth refer to the binary outcome 0 as head and 1 as tail.

To illustrate why composition fails in this setting, let's consider the case of tossing a 4-sided coin among $n = 3k + 1$ where $n_0 = 1$ and $n_1 = n_2 = n_3 = k$ for $k \geq 2$. A strawman solution is as follows:

- First, 0- and 1-supporters together act as a group G_{head} while the 2- and 3-supporters together act a group G_{tail}. These two groups G_{head} and G_{tail} act as the head-supporters and the tail-supporters, respectively, and run an instance of $\Pi(k + 1, 2k)$ to decide a winner.

- If the group G_{head} wins, then the 0-supporters and 1-supporters run another instance of $\Pi(1, k)$. If the output of this coin-toss is head, then the output is 0, otherwise it is 1. Conversely, if G_{tail} wins, then we toss a binary coin among the 3- and 4-supporters, and output 3 (resp., 4) if the binary coin-toss is a head (resp., tail).

One can easily verify that the output is a uniform 4-sided coin in an honest execution. However, imagine a coalition of size $k + 1$ that controls all 0-supporters and 1-supporters. This coalition behaves honestly in the first instance of $\Pi(k + 1, 2k)$. If group G_{head} wins, they can manipulate the second instance $\Pi(1, k)$ and make it output 1. Consequently, the expected utility of this coalition is $\frac{k}{2}$, which is strictly larger than the honest expected utility $\frac{k+1}{4}$. This attack remains effective regardless of how the initial groups are partitioned.

This example shows that the direct composition of CSP-fair binary coins does not guarantee a fair multi-sided coin-toss that tolerates majority-sized coalitions. On a high level, to guarantee the fairness of the overall output, we require fairness from each individual run of Π, which places a bound on the adversarial coalition's size in each of those runs. In the above example, the composition succeeds only when the adversary controls less than half of G_{head} and G_{tail}, imposing significant constraints on the adversary's coalition budget and spread across supporters. As expected, if the restrictions are lifted, and the coalition is allowed to be arbitrary but within a certain larger threshold, the attack we described above immediately comes into play. The above issue seems to generalize and leave moot the composition approach if we wish to tolerate a dishonest majority for multi-sided coin-tosses. In the hopes of obtaining better parameters, we will then resort to building a CSP-fair multi-sided coin-toss from scratch.

2.2 Construction of Uniform Ternary Coin-Toss

One can view our first step as extending the CSP-fair binary coin-toss protocol in [WAS22]. Given a preference profile $[n_0, n_1, n_2]$ where $n_0 \leq n_1 \leq n_2$, we will partition the players into two groups: G_0 that contains all the 0-supporters, and G_{oth} that has all the remaining players. We refer to 1- and 2- supporters as oth-supporters and use n_{oth} to denote $n_1 + n_2$.

At a high level, each group independently decides on a random ternary coin, and the output is the sum of the two coins. To do this, each group $b \in \{0, \mathsf{oth}\}$ will separately run a sub-protocol $\mathsf{GroupToss}^b[\ell, k]$ (described below), where ℓ implies that the output of the sub-protocol should be an ℓ-sided coin, and k is a parameter that we will specify later. For the ternary coin-toss, we set $\ell = 3$. The sub-protocol $\mathsf{GroupToss}^b[\ell, k_b]$ is executed amongst the group G_b and is based on Shamir's secret sharing scheme (We refer the readers to Section 3.2 for a formal definition).

Protocol 2.1 (**Ternary** $\mathsf{GroupToss}^b[\ell, k]$**).**
Sharing Phase.

1. Each b-supporter $i \in G_b$ chooses a random coin $c_i \xleftarrow{\$} \{0, \ldots, \ell - 1\}$ and share it using $(k+1)$-out-of-n_b secret sharing. If a player j aborted, remove it from G_b.
2. Each remaining player $i \in G_b$ computes s_i, the sum of the shares it received from players in G_b.[a]

Reconstruction phase.

1. Players in G_b reconstruct the secret $\widetilde{s}$ by broadcast s_i. If reconstruction succeeded, output $s := \widetilde{s} \mod \ell$. Otherwise, output $\perp$.

[a] Shamir secret sharing satisfies linearity: if every player i has a share s_i of secret s and s'_i of secret s', then k or more players can reconstruct $s + s'$ using shares $s_i + s'_i$.

This sub-protocol satisfies the following properties:

- *Binding:* The sharing phase uniquely binds to a secret s, such that the reconstruction either fails or outputs secret s.
- *Knowledge threshold:* If the coalition controls k or less players, then they have no idea of the secret s at the end of the sharing phase; otherwise, the coalition can control the value of coin s.
- *Liveness:* If the coalition controls $n - k$ or more players, they can fail the reconstruction. Otherwise, the reconstruction must succeed.

With this sub-protocol, we can describe our fair uniform ternary-coin-toss protocol (see Protocol 2.2 below). At a high level, each group G_b for $b \in \{0, \mathsf{oth}\}$ will use $\mathsf{GroupToss}$ to decide a ternary coin s_b. The final outcome will then be the sum of the two coins. Note that when group G_0 runs the $\mathsf{GroupToss}^0[\ell, k_0]$ sub-protocol, group G_{oth} is able to observe the messages in the broadcast channel, although they do not need to send any message in $\mathsf{GroupToss}^0[\ell, k_0]$. We refer to this passive observation as the group G_{oth} *auditing* the $\mathsf{GroupToss}^0[\ell, k_0]$ sub-protocol. Similarly, when

G_{oth} runs the $\mathsf{GroupToss}^{\mathsf{oth}}[\ell, k_{\mathsf{oth}}]$ sub-protocol, group G_0 is the one auditing. Here k_0 and k_{oth} are protocol parameters that will be specified later.

Protocol 2.2 (**Fair uniform ternary coin-toss).**
Sharing phase

1. G_0 runs the sharing phase of $\mathsf{GroupToss}^0[3, k_0]$. Group G_{oth} audits.

2. G_{oth} runs the sharing phase of $\mathsf{GroupToss}^{\mathsf{oth}}[3, k_{\mathsf{oth}}]$. Group G_0 audits.

Reconstruction phase.

1. G_0 runs the reconstruction phase of $\mathsf{GroupToss}^0[3, k_0]$ to reconstruct s_0 while G_{oth} audits. If the reconstruction fails, set $s_0 = 0$.

2. G_{oth} runs the reconstruction phase of $\mathsf{GroupToss}^{\mathsf{oth}}[3, k_{\mathsf{oth}}]$ to reconstruct s_{oth} while G_0 audits. If the reconstruction fails, output $s = 0$ as the *final coin value.* Otherwise, output $(s_0 + s_{\mathsf{oth}}) \mod 3$.

Remark 2.3 (Asymmetry of the protocol)**.** *The protocol is asymmetric in handling failures: if G_0 fails to reconstruct s_0, we set s_0 to be 0; however, if group G_{oth} fails to reconstruct s_{oth}, we directly output $s = 0$. This asymmetry is essential: when G_{oth} reconstructs the coin, the value of s_0 is already publicly known. Thus, to prevent potential exploitation by the coalition based on s_0, we output 0 as a punishment when the reconstruction of s_{oth} fails. This ensures fairness in our coin-toss protocol and is formalized as a crucial property (see Condition 3).*

Threshold parameter selection. Suppose the coalition controls $t = t_0 + t_{\mathsf{oth}}$ number of players, where t_0 and t_{oth} are the number of 0-supporters and other supporters in the coalition, respectively. We aim to select the threshold parameters k_0 and k_{oth}, such that the following three conditions are satisfied. Intuitively, the following three conditions prevent an adversarial coalition from trivially biasing the final output.

1. *The coalition cannot control both coins.*

2. *If the coalition controls the s_{oth} coin, it cannot fail the reconstruction of the coin s_0.* Otherwise, the coalition can always fail the reconstruction of s_0 and make the protocol outputs s_{oth}.

3. *If the coalition can fail the reconstruction of the s_{oth} coin, it must NOT have incentives to bias the output to* 0. If the coalition controls at least $n_{\mathsf{oth}} - k_{\mathsf{oth}}$ number of other supporters such that it can choose to fail the reconstruction of s_{oth} and force the protocol to output 0, then its joint utility must not increase. In other words, the coalition gains utility by forcing the protocol output 0 is t_0, while the honest expected utility of the coalition is $\frac{t}{3}$. Therefore, if $t_{\mathsf{oth}} \geq n_{\mathsf{oth}} - k_{\mathsf{oth}}$, it must be that $t_0 \leq \frac{t}{3}$.

Remark 2.4 (Preferring outcome 0)**.** *In the ternary coin-toss, the coalition does not have incentives to bias the output to 0 when $t_0 \leq \frac{t}{3}$. Note that even when $\frac{t}{3} < t_0 < t_{\mathsf{oth}}$, meaning the coalition controls more* oth*-supporters than 0-supporters, they may still have an incentive to bias the output toward 0 to increase utility. This is quite different from the binary coin-toss, where as long as $t_0 < t_{\mathsf{oth}}$, the coalition is not incentivized to bias the output towards 0. This distinction results in the different landscapes for the CSP-fair multi-sided coin-toss protocols and the binary coin-toss protocols.*

Now, we explain why the above conditions are sufficient for CSP-fairness. First, due to Condition 3, the coalition does not want to fail the reconstruction of s_{oth}. There are two cases to consider. 1.) If $t_{\mathsf{oth}} \leq k_{\mathsf{oth}}$, i.e., the coalition does not have control over the value of s_{oth}. Then, the output must be uniformly random since s_{oth} is guaranteed to be reconstructed; 2.) If $t_{\mathsf{oth}} > k_{\mathsf{oth}}$, then by Condition 1 and 2, s_0 must be independent from s_{oth}, uniformly random, and successfully reconstructed. Therefore, the output must be uniformly random. Together, this guarantees the CSP-fairness of the ternary coin-toss protocol Protocol 2.2.

Now the problem of finding the maximum size of the coalition that Protocol 2.2 can tolerate boils down to finding the maximum t such that there exists a feasible solution for k_0 and k_{oth} satisfying the above conditions. We give the maximum t and the corresponding k_0 and k_{oth} below:

- If $n_{\mathsf{oth}} \geq \frac{9}{2}n_0$, the maximum t is $\lceil \frac{n_0}{2} \rceil + n_{\mathsf{oth}} - 2n_0$ when $k_0 = \lfloor \frac{n_0}{2} \rfloor$ and $k_{\mathsf{oth}} = n_{\mathsf{oth}} - 2n_0$.
- Otherwise, the maximum t is $\lceil \frac{n_0}{2} \rceil + \lfloor \frac{3}{5}n_{\mathsf{oth}} - \frac{1}{5}n_0 \rfloor$ when picking $k_0 = \lfloor \frac{n_0}{2} \rfloor$ and $k_{\mathsf{oth}} = \lfloor \frac{3}{5}n_{\mathsf{oth}} - \frac{1}{5}n_0 \rfloor$.

Remark 2.5 (Phase transition)**.** *Our protocol shares a similar phase transition structure to the binary coin-toss protocol. The intuition is that when n_{oth} is much larger than n_0, it is almost impossible for a majority-sized coalition to prefer* 0 *(when $t_0 \geq \frac{t}{3}$, see Remark 2.4). For example, if $n_0 = 1$ and $n_{\mathsf{oth}} = 6$, then for a coalition to have incentives to bias the output to* 0*, the size of the coalition is at most* 3*, which is smaller than half of the total players. For such profiles, Condition 3 is automatically satisfied and does not add additional constraints. In this case, t is maximized under Condition 1 and 2. Instead, when n_{oth} is closer to n_0, it is possible that a majority-sized coalition prefers* 0*. In this case, t is maximized under all three conditions.*

Protocol 2.2 can be generalized to arbitrary uniform m-sided coin. We describe this protocol and its proof in Section 4. In **??**, we plot the behavior of t as function of n_0, n_{oth}, and m.

2.3 Construction of Non-Uniform Ternary Coin-Toss

Once we have a protocol for fair, uniform coin-tosses, we can expect to have a simple compiler that maps a uniform coin-toss protocol to a non-uniform one. Consider the following example:

(Eg.1) The desired distribution $\mathcal{D} = \{\frac{1}{5}, \frac{3}{5}, \frac{1}{5}\}$, and preference profile $n_0 = 4, n_1 = 5$, and $n_2 = 8$.

The natural idea is to let the players run the sharing phase and reconstruction phase of Protocol 2.2 for 5-sided coins and then map the output to a ternary coin based on the desired distribution.

Protocol 2.6 (**Strawman solution: fair non-uniform ternary coin-toss**)**.**
Sharing phase

1. G_0 (G_{oth}, resp.) runs the sharing phase of $\mathsf{GroupToss}^0[5, k_0]$ (resp. $\mathsf{GroupToss}^{\mathsf{oth}}[5, k_{\mathsf{oth}}]$). The other group audits.

Reconstruction phase.

1. G_0 runs the reconstruction of $\mathsf{GroupToss}^0[5, k_0]$ to compute s_0. If failed, set $s_0 = 0$.
2. G_{oth} runs the reconstruction of $\mathsf{GroupToss}^{\mathsf{oth}}[5, k_{\mathsf{oth}}]$ to computer s_{oth}. If failed, output

$s = 0$. Otherwise, compute $\widetilde{s} = s_0 + s_{\mathsf{oth}} \mod 5$. If $\widetilde{s} = 0$, output $s = 0$; if $\widetilde{s} = 4$, output $s = 2$; otherwise, output $s = 1$.

Clearly, when everyone behaves honestly, we get a ternary coin following the desired distribution $\mathcal{D}$. However, this protocol does not always tolerate majority-sized coalitions. Consider a coalition $\mathcal{A}$ of majority size 9 that controls 2 number of 0-supporters and 7 number of 2-supporters. The expected honest utility of this coalition is $\frac{9}{5}$. Given the preference profile, if we choose k_0 and k_{oth} such that Condition 1 and Condition 2 both hold, then it must be that $k_{\mathsf{oth}} \geq 7$. Therefore, $\mathcal{A}$ can fail the reconstruction of s_{oth} and bias the output towards 0. In this case, the utility will be 2, which is strictly higher than the expected honest utility!

This example implies that, for non-uniform distribution, the *least preferred outcome* does not just depend on the number of supporters. Instead, it also depends on the desired distribution $\mathcal{D}$. To see this, recall that one crucial condition we need is that when the coalition fails s_{oth}, it must *not* have incentives to bias the output to 0. In the case of arbitrary distribution, this means

$3'$ *If the coalition can fail s_{oth} coin's reconstruction, it must NOT have incentives to bias the output to* 0. If the coalition controls at least $n_{\mathsf{oth}} - k_{\mathsf{oth}}$ number of other supporters such that it can force the output to 0, then its joint utility must not increase. Alternatively, if $t_{\mathsf{oth}} \geq n_{\mathsf{oth}} - k_{\mathsf{oth}}$, then $t_0 \leq \sum p_j t_j$.

In the worst case, if $t_0 \leq \frac{\min\{p_1,p_2\}}{1-p_0} \cdot t_{\mathsf{oth}}$, this would ensure $t_0 \leq \sum p_j t_j$. In the above example (Eg.1), $\frac{\min\{p_1,p_2\}}{1-p_0} = \frac{1}{4}$. Thus, we need $t_0 \leq \frac{t_{\mathsf{oth}}}{4}$ to guarantee that the coalition is *not* incentivized to bias the output to 0. This puts a rather restricted constraint on the coalition size the protocol can tolerate. Therefore, we need to redefine the *least preferred outcome*, denoted by lpo. In line with this, we define a *balancing parameter* p_j^* for each outcome j as $p_j^* = \frac{\min_{j' \neq j}\{p_{j'}\}}{1-p_j}$.

Basically, p_j^* measures the maximum fraction of $\frac{t_j}{t-t_j}$, where t_j is the number of j-supporters the coalition can control, such that the coalition never wants to bias the output towards j. It is easy to see that for any distribution, $p_j^* \leq \frac{1}{m-1}$, and for uniform distribution, $p_j^* = \frac{1}{m-1}$ for all j.

Intuitively, we want to choose the *least preferred outcome* to be the one with a small number of supporters but a large balancing parameter. Specifically, we want to make sure that $p_j^*(n-n_j) \geq n_j$ for the least preferred outcome j. The larger the gap, the easier it is for a majority-sized coalition to satisfy Condition $3'$ Therefore, we define the least preferred outcome as $\mathsf{lpo} := \arg\max_j (p_j^*(n-n_j) - n_j)$. For the example we considered earlier for the distribution $\mathcal{D} = \{\frac{1}{5}, \frac{3}{5}, \frac{1}{5}\}$, we have $p_0^* := p_2^* := \frac{1}{4}$ and $p_1^* := \frac{1}{2}$. The lpo in this case is $j = 1$, contrary to the uniform case, which would adjudge 0 as the least preferred outcome.

We note that for the uniform distribution, the above definition for the least preferred outcome collapses to the outcome with the least number of supporters. With this definition, we can modify Protocol 2.6 by making lpo-supporters act as group G_0 and all other players as group G_{oth}. Moreover, if s_{oth} fails, the protocol outputs lpo as the punishment. One can verify that the least preferred outcome in the example (Eg.1) above will be 1. With the above modification, Protocol 2.6 is able to tolerate any 9-sized coalitions.

To generalize to arbitrary distributions $\mathcal{D}$, we can approximate the distribution using U_{2^δ}, where δ is the correctness parameter. The protocol works as follows: Given a distribution $\mathcal{D}$ of an m-sided coin and the preference profile, let lpo be the least preferred outcome. Let G_0 denote all the lpo-supporters, and G_{oth} denote other players. If G_{oth} fails to reconstruct s_{oth}, the protocol outputs lpo as punishment.

By a similar argument as for the uniform coin-toss, as long as there exist solutions to k_0, k_{oth}, and t such that Condition 1, 2, and $3'$ are all satisfied, the resulting protocol is CSP-fair against

t-sized coalitions. Through optimization, our protocol satisfies CSP-fairness against any t-sized coalition, where

$$t = \begin{cases} \left\lceil \frac{n_{\mathsf{lpo}}}{2} \right\rceil + n_{\mathsf{oth}} - \frac{n_{\mathsf{lpo}}}{p^*}, & \text{if } n - n_{\mathsf{lpo}} \geq \frac{p^*+4}{2p^*} n_{\mathsf{lpo}} \\ \left\lceil \frac{n_{\mathsf{lpo}}}{2} \right\rceil + \left\lfloor \frac{p^*+1}{p^*+2} n_{\mathsf{oth}} - \frac{n_{\mathsf{lpo}}}{2(p^*+2)} \right\rfloor, & \text{otherwise,} \end{cases} \tag{3}$$

where n_{lpo} is the number of lpo-supporters, while $n_{\mathsf{oth}} = n - n_{\mathsf{lpo}}$ is the number of all other players. The formal protocol and the proofs are given in Section 5.

Upgrade to malicious security. Our protocol can be generalized to achieve malicious security against the same-sized coalition as in (3). The formal description of the protocol and the proofs are given in Section 7.

2.4 Impossibility Result

Our impossibility result has two steps; while it is inspired by [CGL+18a] and [WAS22], it requires significantly new techniques. In the first step, we transform any n-party coin-toss protocol Π into a *three*-party protocol Π_3 by partitioning the n parties into three supernodes $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ such that Π_3 satisfies:

(C1) **Lone-wolf condition**: $\mathcal{S}_1$ (or $\mathcal{S}_3$) alone cannot bias the output towards *any* direction.

(C2) **Wolf-minion condition**: $\mathcal{S}_1$ and $\mathcal{S}_2$ (or $\mathcal{S}_3$ and $\mathcal{S}_2$) cannot decrease the probability of getting a 0.

(C3) **T_2-equity condition**: for all but a negligible fraction of $\mathcal{S}_2$'s randomness T_2, the distribution of $f(T_2)$ is indistinguishable from the uniform distribution, where $f(T_2)$ is the random variable denoting the outcome in an honest execution where $\mathcal{S}_2$'s randomness is fixed to T_2.

Crucially, our wolf-minion condition is inherently different from that in previous works. The difference and the reason behind it will be clear after we explain our techniques. Then, we show that no coin-toss protocol can satisfy the above three conditions simultaneously. To show this, we generalize Cleve's impossibility proof to the multi-sided coin in Section 8.2, which may be of independent interest.

In the following, we will only focus on the first step, and in particular on how to partition the n players into three supernodes $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{S}_3$. Without loss of generality, assume that $n_0 \leq n_1 \leq n_2$. Supernode $\mathcal{S}_1$ and $\mathcal{S}_3$ each contains α_j many j-supporters, where $\alpha_j \leq \frac{n_j}{2}$, and $\mathcal{S}_2$ contains the remaining players. Each supernode emulates the execution of all players in it. The messages between players in the same supernode are dealt with internally, and those between players in different supernodes are treated as messages between these supernodes. Thus, Π can naturally be translated into a three-party protocol Π_3 among $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$.

We now explain why Π_3 must satisfy the above three conditions with a concrete example (Eg.2). Then, we give the constraints that t and α_j (for $j = 0, 1, 2$) need to satisfy for the three conditions hold in general.

(Eg.2) For preference profile $n_0 = 2, n_1 = 4$, and $n_2 = 5$, we will show that no protocol is CSP-fair against coalitions of size $t = 9$. As a comparison, our protocol tolerates any coalition of size up to 6.

The partition for (Eg.2) is as follows: $\alpha_0 = \alpha_1 = \alpha_2 = 1$. Next, we show that under such partitions, Π_3 must satisfy all three conditions, assuming Π is fair against any coalition of size 9.

- *Lone-wolf condition*: For the sake of contradiction, suppose that $\mathcal{S}_1$ can bias towards outcome j. Then $\mathcal{S}_1$ can collude with another j-supporter in $\mathcal{S}_3$. This coalition controls two j-supporters and one j'-supporter for each $j' \neq j$. It can bias the outcome towards j, which contradicts CSP-fairness.

- *Wolf-minion condition*: For the sake of contradiction, assume that $\mathcal{S}_1$ and $\mathcal{S}_2$ together can decrease the probability of getting a 0. Then $\mathcal{S}_1$ and $\mathcal{S}_3$ can collude with the 1-supporter in $\mathcal{S}_3$. This new coalition contains a single 0-supporter, four 1-supporters, and four 2-supporters. In addition, it can increase the probability of getting a 1 or a 2, which contradicts CSP-fairness.

- *T_2-equity condition*: by a similar argument as above for the wolf-minion condition, we can show that for all but negligible fraction of T_2, $f(T_2)$ must not be able to bias towards 1 or 2. Since $\mathbf{E}_{T_2}[f(T_2)]$ is negligibly close to the uniform distribution, it must be that $f(T_2)$ cannot bias towards any direction.

From the arguments above, the main constraints on α_0, α_1, and α_2 are posed by the lone-wolf condition and the wolf-minion condition. To generalize the above argument, we require that α_0, α_1, and α_2 satisfy:

(Ctr-1) *Non-negativity*: $0 \leq \alpha_j \leq \frac{n_j}{2}$ for $j = 0, 1, 2$.

(Ctr-2) *Lone-wolf condition*: $\alpha_j \leq n_0 - 1$. For the condition to hold, we must be able to construct for any j, a coalition $\mathcal{A}_j$ that contains $\mathcal{S}_1$ and some other players, such that $\mathcal{A}_j$ prefers j. Thus, α_j must be smaller than n_0.

(Ctr-3) *Wolf-minion condition*: $n_1 \geq n_2 - 2\alpha_2$. This is necessary to construct a coalition $\mathcal{A}$ that contains $\mathcal{S}_1$, $\mathcal{S}_2$, and some other 1-supporters such that *the number of* 1- *and* 2-*supporters controlled by the coalition are the same.* For such coalitions, if it is able to increase the probability of 1 or 2, it must be able to increase its expected utility. The size of such a coalition $\mathcal{A}$ is $n_0 - \alpha_0 + 2(n_2 - \alpha_2)$.

The above constraints imply that for $n_1 \geq n_2 - \alpha_2$, no CSP-fair protocol can tolerate $n_0 - \alpha_0 + 2(n_2 - \alpha_2)$ corruptions. By picking α_0 and α_2 that minimizes this term, we get an impossibility that no protocol is CSP-fair against t^*-sized coalition where $t^* = \frac{n_0}{2} + \max\{2n_2 - 2n_0 + 2, n_2\}$.

For $n_1 + n_2 \geq \frac{9}{2}n_0$, the gap between the impossibility and the coalition size we can tolerate is $n_2 - n_1 + 2$. Therefore, if $n_2 - n_1$ is a constant, then our protocol is optimal except for an additive constant term.

What if $n_1 < n_2 - \min\{n_0 - 1, \frac{n_2}{2}\}$? For example, what if $n_0 = 2, n_1 = 3$, and $n_2 = 5$? In this case, there is no feasible solution of α_0, α_1, and α_2 such that the above constraints are satisfied. In this case, we consider another version of the wolf-minion condition:

(C2′) **Wolf-minion condition 2**: a semi-malicious non-uniform p.p.t. adversary controlling $\mathcal{S}_1$ and $\mathcal{S}_2$ (or $\mathcal{S}_3$ and $\mathcal{S}_2$) cannot increase the probability of getting a 2.

Now, instead of considering a coalition $\mathcal{A}$ that controls the same number of 1- and 2-supporters, we construct a coalition that controls the same number of 0- and 1-supporters. The corresponding constraint becomes:

(Ctr-3′) *Wolf-minion condition 2*: $n_0 \geq n_1 - \alpha_1$. Only if this is satisfied can we construct a coalition $\mathcal{A}$ that contains $\mathcal{S}_1$ and $\mathcal{S}_2$, as well as some 0-supporters in $\mathcal{S}_3$, such that $\mathcal{A}$ controls the same number of 0- and 1-supporters. This coalition contains $(n_1 - \alpha_1)$ number of 0-supporters, $(n_1 - \alpha_1)$ number of 1-supporters, and at most $(n_2 - \alpha_2 + 1)$ number of 2-supporters.

Together with the constraints (Ctr-1) and (Ctr-2), we get an impossibility result when $n_0 \geq \max\{n_1 - n_0 + 1, \frac{n_1}{2}\}$, or equivalently, when $n_1 \leq 2n_0 - 1$: There is no protocol that is CSP-fair against t^*-sized coalition where $t^* = 2n_1 - 2n_0 + 2 + \max\left\{n_2 - n_0 + 1, \frac{1}{2}n_2\right\}$.

Remark 2.7 (Coalition's structure)**.** *Our constraint for the wolf-minion condition is very different from that of the binary coin-toss [CGL+18a, WAS22], where a coalition of size up to t should not be able to bias the output towards 1 as long as it controls more 1-supporters than 0-supporters. In the case of the ternary coin-toss, imagine a coalition that controls a single 1-supporter and two 2-supporters. It is possible that the coalition has a strategy, under which the distribution of the outcome becomes $\{\frac{1}{2} + \epsilon, 0, \frac{1}{2} - \epsilon\}$ for some positive ϵ. This strategy increases the probability of getting a 2, yet decreases the joint utility of the coalition. To rule out such "bad cases", we only consider coalitions with the following properties: either $t_0 = t_1$, or $t_1 = t_2$, where t_j is the number of j-supporters controlled by the coalition.*

Combining these two cases together, we can prove a lower bound either when n_1 is "relatively close" to n_2, or when n_1 is "relatively close" to n_0: There is no CSP-fair protocol against t^*-sized coalition where

$$t^* = \begin{cases} \frac{n_0}{2} + \max\{2n_2 - 2n_0 + 2, n_2\}, & \text{if } n_1 \geq \max\{n_2 - n_0 + 1, \frac{n_2}{2}\} \\ 2n_1 - 2n_0 + 2 + \max\{n_2 - n_0 + 1, \frac{1}{2}n_2\}, & \text{if } n_1 \leq 2n_0 - 1 \end{cases}$$

Generalization to multi-sided coin. Similar to the impossibility result for ternary coin, we will need the preference profile to satisfy the following property: without loss of generality, assume $n_0 \leq n_1 \leq \cdots \leq n_{m-1}$. There exists a $b \in \{1, \ldots, m-1\}$ such that n_{b-1} is relatively close to n_0, while n_b is relatively close to n_{m-1}. We call this property *b-bipartite.* For b-bipartite preference profiles, we can construct a coalition $\mathcal{A}$ where $t_0 = \cdots = t_{b-1}$, and $t_b = \cdots = t_{m-1}$. We show that if a protocol is CSP-fair against t^*-sized coalition, then the corresponding three-party protocol must satisfy the lone-wolf condition, T_2-equity condition, and the wolf-minion condition that $\mathcal{S}_1$ and $\mathcal{S}_2$ should not be able to increase (or decrease) the probability of getting b or larger outcomes, However, such protocols that satisfy the three conditions simultaneously do not exist. The formal proof is given in Section 8.

3 Preliminaries and Model

Notations. We use m-coin $\mathcal{D}$ to represent the desired distribution of the outcome over the support $\{0, \ldots, m-1\}$, where $\mathcal{D} = \{p_0, \ldots, p_{m-1}\}$. Here $p_j = \Pr[\text{output} = j]$. If $\mathcal{D}$ is a uniform distribution over $\{0, \ldots, m-1\}$, we also call it a uniform m-coin. We use negl to denote a negligible function. Also, we assume that each player is a probabilistic polynomial time (p.p.t.) Turing machine. In our proof, we will rely on the notion of *computationally indistinguishable*: Given two distribution ensembles $\{X^n\}_n$ and $\{Y^n\}_n$, we say that these two ensembles are computationally indistinguishable ($\{X^n\}_n \equiv_c \{Y^n\}_n$) if for any p.p.t. adversary $\mathcal{A}$, we have $|\Pr[\mathcal{A}(x) = 0 : x \overset{\$}{\leftarrow} X^n] - \Pr[\mathcal{A}(x) = 0 : x \overset{\$}{\leftarrow} Y^n]| \leq \mathsf{negl}(n)$.

The boldface letters represent vectors. We use $\{0, \ldots, \ell - 1\}^\theta$ to denote a vector of length θ, where each entry belongs to set $\{0, \ldots, \ell - 1\}$. We use $[a, b]$ to denote all integers from a to b, both sides included. Unless otherwise noted, the sum of two integers is operated over real numbers instead of finite fields. In addition, $\oplus$ denotes bit-wise XOR between two binary vectors.

3.1 Coin-Toss Protocol

In an n-party coin-toss protocol, n players communicate with each other through a pairwise private channel as well as a public broadcast channel. The communication channels are authenticated, i.e., all messages carry the identity of the true sender. In this paper, we assume that the network is synchronous and the protocol proceeds in rounds. The coalition A (also called the adversary in the cryptography analysis sections) can perform *rushing attacks*: in any round r, it waits for honest players to send messages and then decides what messages to send by players in A for round r. At the end of the protocol, the coin outcome is a deterministic, polynomial-time computable function based on the broadcast history, i.e., all messages that have been posted to the broadcast channel.

Correctness. Consider a protocol that tosses an m-coin $\mathcal{D}$. The correctness of the protocol is defined with respect to a correctness parameter δ. A protocol is δ-correct if for any $j \in \{0, \ldots, m-1\}$, the following holds for some negligible function $\mathsf{negl}(\cdot)$.

$$|\Pr[\text{outcome of the protocol is } j] - p_j| \le \mathsf{negl}(\delta).$$

Public preference profile. In a protocol tossing an m-coin $\mathcal{D}$, we assume that each player has a publicly stated preference for *one* outcome in $\{0, \ldots, m-1\}$. The vector of all players' preferences is referred to as the *preference profile* $[n_0, \ldots, n_{m-1}]$, where n_j denotes the number of users who prefer outcome j. We call a player j-supporter if it prefers the outcome j. We use $\mathcal{P}_j$ to denote the set of j-supporters and $\mathcal{P}_{\bar{j}}$ to denote the set of all other players not preferring j.

If there exists an outcome that no player prefers, we call the preference profile as *degenerate preference profile.* Otherwise, we call it a *non degenerate preference profile* if each outcome in $\{0, \ldots, m-1\}$ has at least one supporter.

Utility and strategies. A player's utility is defined as follows: If the outcome agrees with the players' preference, it gets utility 1. Otherwise, it gets 0. A coalition A's utility is the sum of the utilities of all coalition members.

A *malicious* coalition can perform the following types of deviations.

- The coalition can program its randomness used in each round after observing the honest messages of that round;
- The coalition can abort the protocol in some round r, after observing the honest round-r messages. Once a player aborted in round r, it does not participate in future rounds.
- The coalition can send arbitrary messages in each round after observing the honest messages of that round.

We say that a coalition is *semi-malicious* if it is restricted to the first two types of deviations.

CSP-fairness. CSP-fairness states that no coalition can increase its own expected utility except for a negligible term, no matter how it deviates from the protocol. The fairness is parameterized with the security parameter λ. For a coalition A, let $\mathsf{util}_A(S_A, H_{-A})$ denote the expected utility of A when the coalition adopts strategy S_A, while the players not in A follow the protocol honestly. Similarly, let $\mathsf{util}_A(H_A, H_{-A})$ denote the expected utility of A when every player follows the honest strategy, i.e., follows the protocol.

Definition 3.1. *A protocol tossing an m-coin $\mathcal{D}$ for a preference profile P is* cooperative-strategy-proofness *(CSP-fairness) against any t-sized coalition if for any coalition A of size no more than t, for any non-uniform probabilistic polynomial-time (p.p.t.) strategy S_A that A adopts, there exists a negligible function $\mathsf{negl}(\cdot)$, such that $\mathsf{util}_A(S_A, H_{-A}) \leq \mathsf{util}_A(H_A, H_{-A}) + \mathsf{negl}(\lambda)$.*

In this paper, when we say fairness, we mean CSP-fairness.

3.2 Shamir Secret Sharing

The construction of our protocol relies on Shamir's secret sharing. A k-out-of-n secret sharing scheme consists of two algorithms Share and $\mathsf{Reconstruct}$

1. $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s)$: The sharing algorithm takes as input a secret s, and outputs n shares of the secret.
2. $s \leftarrow \mathsf{Reconstruct}(I, (s_i)_{i\in I})$: The reconstruction algorithm takes as input $|I|$ secret shares, and reconstructs the secret only if $|I| \geq k$. Otherwise, the algorithm returns $\perp$.

The t-out-of-n Shamir secret sharing scheme satisfies the following properties:

1. **Correctness**: For any secret s and any sets $I \subset \{1, \ldots, n\}$ such that $|I| \geq k$, the following holds:
$$\Pr[\mathsf{Reconstruct}(I, (s_i)_{i\in I}) = s : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s)] = 1.$$
2. **Security**: For any secrets s and s', for any sets I such that $I \subset \{1, \ldots, n\}$, and $|I| \leq k - 1$, the following holds:
$$\{(s_i)_{i\in I} : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s)\} \equiv_c \{(s'_i)_{i\in I} : (s'_1, \ldots, s'_n) \leftarrow \mathsf{Share}(s')\}.$$
Informally, this means that if a coalition has any $k - 1$ shares, they have no information about the secret.
3. **Linearity**: For any secrets s and s', for any sets $I \subset \{1, \ldots, n\}$ such that $|I| \geq k$, the following holds:
$$\Pr[\mathsf{Reconstruct}(I, (s_i + s'_i)_{i\in I}) = s + s' : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s), (s'_1, \ldots, s'_n) \leftarrow \mathsf{Share}(s')] = 1.$$
Roughly speaking, this means that one can reconstruct $s + s'$ using the sum of the shares of s and s'.

Specifically, Shamir's secret sharing works on a finite field $\mathbb{F}_q$ for some prime q. In the sharing algorithm, it randomly chooses a degree $k-1$ polynomial $f(\cdot)$ such that $f(0) = s$. For $i = \{1, \ldots, n\}$, the share $s_i = f(i)$. The reconstruction algorithm simply takes k or more shares and performs polynomial interpolation to reconstruct $f(0)$.

4 Fair Uniform Coin-Toss: Semi-Malicious Security

In this section, we will give a fair, uniform m-sided coin-toss protocol against a semi-malicious coalition, which is similar to the warmup protocol Protocol 2.2 given in Section 2. We will mainly focus on the case where each outcome has at least one supporter, which we refer to as a non-degenerate preference profile. For the case where there exists an outcome j that is unfavorable to any player, we call it a degenerate preference profile. Since there is a simple protocol that achieves game-theoretic fairness for degenerate preference profiles, we will leave that to the end of this section.

4.1 Sub-Protocol $\mathsf{GroupToss}$

Without loss of generality, we assume that 0 is the least preferred outcome, i.e., $1 \le n_0 \le n_j$ for $j = 1, \ldots, m-1$. Let G_b denote the group of b-supporters for $b \in \{0, \mathsf{oth}\}$, which represents the group of 0-supporters and the group of other players, respectively. Also, we use n_b for $b \in \{0, \mathsf{oth}\}$ to denote n_0 and $n_{\mathsf{oth}} = n - n_0$. The fair coin-toss protocol runs two instances of a $\mathsf{GroupToss}$ sub-protocol: first among the 0-supporters while other players audit, and then among the oth-supporters while 0-supporters audit. Roughly speaking, the $\mathsf{GroupToss}$ sub-protocol decides a random coin for the group of players who invoke this sub-protocol. The final coin would be the sum of the outcomes of the two instances of $\mathsf{GroupToss}$ sub-protocols.

Henceforth, we use $\mathsf{GroupToss}^b$ to denote a sub-protocol among group G_b for $b \in \{0, \mathsf{oth}\}$. The sub-protocol $\mathsf{GroupToss}^b$ is parameterized with two parameters, ℓ and k, where ℓ denotes the number of sides of the coin decided by the sub-protocol and k is the threshold of the secret sharing scheme used in the protocol. We will specify how to choose the parameters later. Below we give a full description of $\mathsf{GroupToss}^b[\ell, k]$ sub-protocol.

Protocol 4.1 ($\mathsf{GroupToss}^b[\ell, k]$ **sub-protocol (semi-malicious version)).**
Sharing Phase. Initialize G_b to be the set of all b-supporters.

1. Each b-supporter $i \in G_b$ chooses a random coin $c_i \overset{\$}{\leftarrow} \{0, \ldots, \ell - 1\}$. It then uses $(k+1)$-out-of-n Shamir secret sharing over $\mathbb{F}_q$ for some $q > m \cdot n$. to split the coin c_i into n_b shares, denoted $\{[c_i]_j\}_{j \in G_b}$, respectively. For each $j \in G_b$, player i sends the j-th share $[c_i]_j$ to player $j \in G_b$. over a private channel.
2. If a player j aborted, remove it from G_b.
3. Each remaining player $i \in G_b$ computes $s_i := \sum_{j \in G_b} [c_j]_i$, where $[c_j]_i$ is the share player i has received from player j.

Reconstruction phase.

1. Every player $i \in G_b$ broadcasts the reconstruction message (i, s_i).
2. If at least $k+1$ number reconstruction message were posted, then reconstruct the secret $\widetilde{s}$ using shares s_i. Output $s = \widetilde{s} \mod \ell$.
3. Otherwise, output $\perp$.

The $\mathsf{GroupToss}$ sub-protocol is an extension of the sub-protocol proposed in [WAS22], and it satisfies the following properties:

- *Uniform randomness.* If everyone behaves honestly, the output is a uniformly random coin over $\{0, \ldots, \ell - 1\}$.
- *Binding.* The sharing phase uniquely binds to a secret s, such that the reconstruction phase either output s or $\perp$.
- *Knowledge threshold.* If at least $k+1$ number of b-supporters are in the coalition, then they can control the outcome s.

  On the other hand, if the coalition controls at most k number of b-supporters, then the coalition's view at the end of the sharing phase is independent of the coin value s that the sharing binds to.

- *Liveness threshold.* If the coalition controls at least $n_b - k$ number of b-supporters, it can cause the reconstruction to output $\perp$. On the other hand, if the coalition controls fewer than $n_b - k$ number of b-supporters, the reconstruction phase must succeed.

Lemma 4.2. *The sub-protocol* $\mathsf{GroupToss}^b[\ell, k]$ *satisfies uniform randomness, binding, knowledge threshold, and liveness threshold properties.*

Proof. The uniform randomness follows from the fact that the reconstructed $\widetilde{s} = \sum_{j \in G_b} c_j$ by the linearity of Shamir secret sharing. Since the field size is $q > m \cdot n$, the final output $s \mod \ell = \sum_{j \in G_b} c_j \mod \ell$. Had every player behaved honestly, $\widetilde{s} \mod \ell$ is a uniformly random coin over $\{0, \ldots, \ell - 1\}$. The binding property follows from the correctness of Shamir's secret sharing and the fact that the coalition adopts semi-malicious strategies.

For the knowledge threshold property, note that if the coalition controls at least $k+1$ number of b-supporters, then they will know the value of c_j for every honest player j during the sharing phase. Therefore, they can choose the coalition's coin value accordingly to program the outcome. On the other hand, if the coalition controls at most k number of b-supporters, then their view at the end of the sharing phase is independent of the coin value by the security of the secret sharing scheme. The liveness threshold property also follows from the security of secret sharing. □

4.2 CSP-fair Uniform Coin-Toss

To toss a uniform m-coin, our fair coin-toss protocol first runs $\mathsf{GroupToss}^0[m, k_0]$ among the group of 0-supporters G_0, while other players in G_{oth} can observe the messages posted in the broadcast channel but not send messages. We say that G_{oth} *audits* when G_0 runs $\mathsf{GroupToss}^0[m, k_0]$. Similarly, G_{oth} then runs $\mathsf{GroupToss}^{\mathsf{oth}}[m, k_{\mathsf{oth}}]$ among other supporters while the 0-supporters audit. We will later specify how to choose the threshold parameters k_0 and k_{oth} and how they affect the size of the coalition we can tolerate.

Protocol 4.3 (Fair uniform m coin-toss with semi-malicious security).
Input: A non-degenerate preference profile $[n_0, \ldots, n_{m-1}]$ where 0 is the least preferred outcome.
Sharing phase

1. G_0 runs the sharing phase of $\mathsf{GroupToss}^0[m, k_0]$. Other players audit.
2. G_{oth} run the sharing phase of $\mathsf{GroupToss}^{\mathsf{oth}}[m, k_{\mathsf{oth}}]$. The 0-supporters audit.

Reconstruction phase.

1. G_0 runs the reconstruction phase of $\mathsf{GroupToss}^0[m, k_0]$. Other players audit. If the reconstruction succeeded, let its outcome be s_0. Otherwise, set $s_0 = 0$.
2. G_{oth} runs the reconstruction phase of $\mathsf{GroupToss}^{\mathsf{oth}}[m, k_{\mathsf{oth}}]$. The 0-supporters audit. If the reconstruction fails, output $s = 0$ as the *final coin value.* Otherwise, let its outcome be s_{oth} and output $(s_0 + s_{\mathsf{oth}}) \mod m$ as the final coin value.

Threshold parameter selection. Suppose the coalition controls $t = t_0 + t_{\mathsf{oth}}$ number of players, where t_0 and t_{oth} are the number of 0-supporters and other supporters in the coalition, respectively. Our goal is to select the threshold parameters k_0 and k_{oth}, such that the following three conditions are satisfied:

(C1) *The coalition cannot control both coins.* If the coalition controls at least $k_0 + 1$ number of 0-supporters, the coalition must control at most k_{oth} number of other supporters due to the corruption budget t, and vise versa.

(C2) *If the coalition controls the s_{oth} coin, it cannot fail the reconstruction of the coin s_0.* If the coalition controls at least $k_{\mathsf{oth}} + 1$ number of other supporters, then due to its corruption budget, it can control at most $n_0 - k_0 - 1$ number of 0-supporters. Otherwise, the coalition can fail the reconstruction of s_0, and make the protocol outputs the value s_{oth} they choose.

(C3) *If the coalition can fail the reconstruction of the s_{oth} coin, it must NOT prefer* 0. If the coalition controls at least $n_{\mathsf{oth}} - k_{\mathsf{oth}}$ number of other supporters such that it can fail the reconstruction of s_{oth} and force the protocol to output 0, then its joint utility must not increase. In other words, the utility that the coalition gains by forcing the protocol output 0 is t_0, while the honest expected utility of the coalition is $\frac{t}{m}$. This guarantees that a coalition is not incentivized to fail the reconstruction of s_{oth}.

We first show that the above three conditions are enough to guarantee that the protocol satisfies CSP-fairness.

Lemma 4.4. *Given a preference profile $[n_0, \ldots, n_{m-1}]$ and a size of the coalition t. Suppose that the parameters k_0 and k_{oth} are chosen such that conditions (C1), (C2), (C3) are all satisfied, then Protocol 4.3 satisfies CSP-fairness against any semi-malicious coalition of size no more than t.*

Proof. Due to condition (C3), it never makes sense for the coalition to fail the reconstruction of s_{oth}. Since if they fail the reconstruction of s_{oth}, they get a utility $t_0 \leq \frac{t}{m}$ by condition (C3). Thus, we may assume that s_{oth} is always successfully reconstructed. There are two cases:

If $t_{\mathsf{oth}} \leq k_{\mathsf{oth}}$: the value of s_{oth} is uniform and independent of the coalition's view at the end of the sharing phase. Since s_{oth} will be successfully reconstructed, the final outcome must be uniformly random.

If $t_{\mathsf{oth}} > k_{\mathsf{oth}}$: the coalition can control the value of s_{oth}. By conditions (C1) and (C2), it must be that s_0 is uniform and independent of the coalition's view at the end of the sharing phase, and s_0 is guaranteed to be reconstructed. In this case, the final outcome $s_0 + s_{\mathsf{oth}}$ must also be uniformly random. □

According to Lemma 4.4, we only need to care about choosing the parameters k_0 and k_{oth} such that the above three conditions are satisfied. Next, we show that if the parameters k_0 and k_{oth} satisfy the following constraints, then they satisfy the above three conditions.

Parameter Constraints 4.5 (uniform m-coin, semi-malicious version)**.**

- $0 \leq k_0 \leq n_0$, $0 \leq k_{\mathsf{oth}} \leq n_{\mathsf{oth}}$,
- $t \leq k_0 + k_{\mathsf{oth}} + 1$,
- $t \leq n_0 - k_0 + k_{\mathsf{oth}}$,
- If $n_{\mathsf{oth}} - k_{\mathsf{oth}} < (m-1)n_0$, then $t \leq \frac{m}{m-1}(n_{\mathsf{oth}} - k_{\mathsf{oth}})$.

Lemma 4.6. *If k_0, k_{oth} and t satisfy Parameter Constraints 4.5, then it satisfies conditions (C1), (C2) and (C3).*

Proof. The first constraint is to guarantee the feasibility. The second and third constraints directly correspond to condition (C1) and (C2). In this proof, we mainly focus on proving that the last constraint implies condition (C3). Suppose that the coalition is able to fail the reconstruction of the coin s_{oth}, i.e., $t_{\mathsf{oth}} \geq n_{\mathsf{oth}} - k_{\mathsf{oth}}$. There are two possible cases:

Case 1: If $n_{\mathsf{oth}} - k_{\mathsf{oth}} \geq (m-1)n_0$. Then $t_{\mathsf{oth}} \geq (m-1)n_0$. Since the number of 0-supporters in the coalition must satisfy $t_0 \leq n_0$, we have $t_0 \leq \frac{1}{m-1} \cdot t_{\mathsf{oth}}$. This is equivalent to $t_0 \leq \frac{t}{m}$. Thus, condition (C3) is automatically satisfied under this case.

Case 2: If $n_{\mathsf{oth}} - k_{\mathsf{oth}} < (m-1)n_0$, then we have $t \leq \frac{m}{m-1}(n_{\mathsf{oth}} - k_{\mathsf{oth}}) \leq \frac{m}{m-1} t_{\mathsf{oth}}$ by the last constraint in Parameter Constraints 4.5. This is equivalent to $t_0 = t - t_{\mathsf{oth}} \leq \frac{1}{m-1} \cdot t_{\mathsf{oth}}$. By the same reasoning as in Case 1, we have that condition (C3) is satisfied. □

Optimal resilience for Protocol 4.3. Given a preference profile $[n_0, \ldots, n_{m-1}]$, finding the maximum size of the coalition that the protocol can tolerate is equivalent to solving for the maximum t such that there exists a feasible solution for k_0 and k_{oth} satisfying Parameter Constraints 4.5. Below, we give the optimal resilience for Protocol 4.3. Given a preference profile $[n_0, \ldots, n_{m-1}]$,

Case	k_0	k_{oth}	t
If $n_{\mathsf{oth}} \geq \frac{4m-3}{2} n_0$	$\lfloor \frac{n_0}{2} \rfloor$	$n_{\mathsf{oth}} - (m-1)n_0$	$\lceil \frac{n_0}{2} \rceil + n_{\mathsf{oth}} - (m-1)n_0$
Otherwise	$\lfloor \frac{n_0}{2} \rfloor$	$\lfloor \frac{m}{2m-1} n_{\mathsf{oth}} - \frac{m-1}{2(2m-1)} n_0 \rfloor$	$\lceil \frac{n_0}{2} \rceil + \lfloor \frac{m}{2m-1} n_{\mathsf{oth}} - \frac{m-1}{2(2m-1)} n_0 \rfloor$

Table 1: Optimal resilience of Protocol 4.3 against semi-malicious adversary.

One can easily check from the table above that the size of the coalition we can tolerate is at least $\frac{n}{2}$, except that for a few cases for which we can tolerate $\lfloor \frac{n}{2} \rfloor$[2]. It remains to prove that the parameters given in Table 1 indeed give the maximum t we can tolerate given the parameter constraints 4.5.

Lemma 4.7. *Assuming $n_{\mathsf{oth}} \geq (m-1)n_0$ and $n_0 \geq 1$. For the constraints given in 4.5, the corruption budget t is maximized when k_0 and k_{oth} are chosen as in Table 1.*

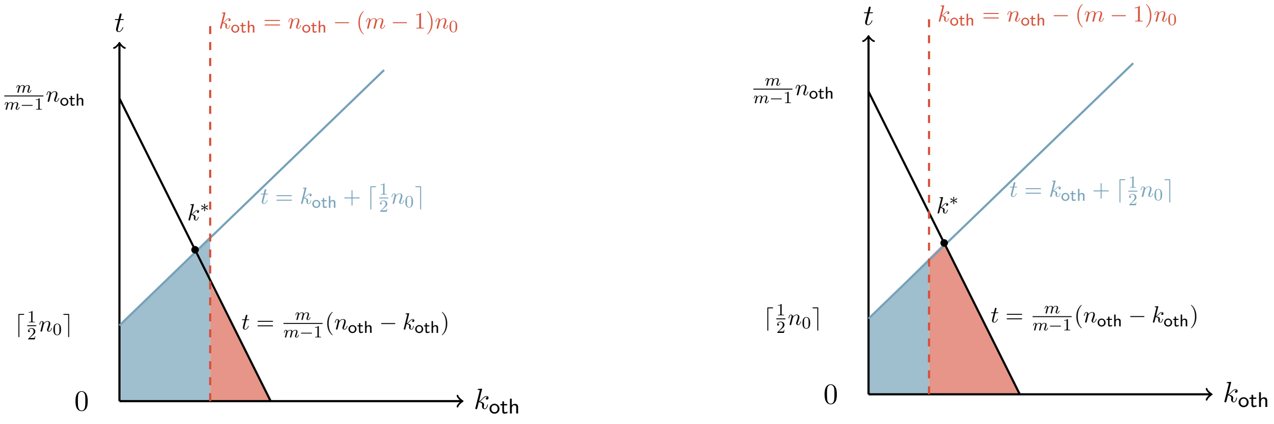
Proof. Note that k_0 only appears in the conditions (C1) and (C2), for any fixed k_{oth}, t is maxmized when $k_0 + k_{\mathsf{oth}} + 1 = n_0 - k_0 + k_{\mathsf{oth}}$. After rounding, we get that t is maximized when $k_0 = \lfloor \frac{n_0}{2} \rfloor$ for any fixed k_{oth}.

Plugging $k_0 = \lfloor \frac{n_0}{2} \rfloor$ back to the system, the problem now boils down to finding k_{oth} that maximizes t, subject to the following constraints

$$
\begin{aligned}
& t \leq \left\lceil \frac{n_0}{2} \right\rceil + k_{\mathsf{oth}}; \\
& \text{If } n_{\mathsf{oth}} - k_{\mathsf{oth}} < (m-1)n_0, \text{ then } t \leq \frac{m}{m-1}(n_{\mathsf{oth}} - k_{\mathsf{oth}}).
\end{aligned}
\tag{4}
$$

Note that the two straight lines $t = \lceil \frac{n_0}{2} \rceil + k_{\mathsf{oth}}$ and $t = \frac{m}{m-1}(n_{\mathsf{oth}} - k_{\mathsf{oth}})$ intersects at $k_{\mathsf{oth}} = k^* = \frac{m}{2m-1} n_{\mathsf{oth}} - \frac{m-1}{2m-1} \lceil \frac{n_0}{2} \rceil$, as denoted in Figure 1. We now separate the rest of the optimization into two cases.

[2] When n_0 is even and n_{oth} is odd, and moreover, $n_{\mathsf{oth}} < (m-1)n_0 + (2m-1)$, the size of the coalition we can tolerate is $\lfloor \frac{n}{2} \rfloor$ due to the rounding. This holds also for the binary coin-toss, in which [WAS22] showed that this is the optimal we can achieve.



(a) The case where $n_{\mathsf{oth}} \geq \frac{4m-3}{2}n_0$.

(b) The case where $n_{\mathsf{oth}} < \frac{p^*+4}{2p^*}n_0$.

Figure 1: Feasible region (filled area) defined by constraints (4). The blue area represents the feasible region if $n_{\mathsf{oth}} - k_{\mathsf{oth}} \geq (m-1)n_0$; while the red area represents the feasible region if $n_{\mathsf{oth}} - k_{\mathsf{oth}} < (m-1)n_0$.

Case 1: If $n_{\mathsf{oth}} \geq \frac{4m-3}{2}n_0$, then $n_{\mathsf{oth}} - (m-1)n_0 \geq k^*$, and the feasible region is depicted in Figure 1a. From the figure we can see that t is maximized when $k_{\mathsf{oth}} = n_{\mathsf{oth}} - (m-1)n_0$. This gives $t = \frac{m}{m-1}((m-1)n_0 - 1)$.

Case 2: If $n_{\mathsf{oth}} < \frac{4m-3}{2}n_0$, then $n_{\mathsf{oth}} - (m-1)n_0 < k^*$, and the feasible region is depicted in Figure 1b. From the figure, one can see that t is maximized when picking $k_{\mathsf{oth}} = \lfloor k^* \rfloor$. In this case, $t = \lfloor k^* \rfloor + \lceil \frac{n_0}{2} \rceil = \lfloor \frac{m}{2m-1}n_{\mathsf{oth}} - \frac{m-1}{2m-1}n_0 \rfloor + \lceil \frac{n_0}{2} \rceil$. □

Theorem 4.8. *Protocol 4.3 is a CSP-fair coin-toss protocol for tossing a uniform m-coin for a preference profile $[n_0, \ldots, n_{m-1}]$ where $n_j \geq n_0 \geq 1$ for any $j \in \{0, \ldots, m-1\}$. It can tolerate any coalition of size no more than*

$$t = \begin{cases} \lceil \frac{n_0}{2} \rceil + n - m \cdot n_0, & \text{if } n \geq \frac{4m-1}{2}n_0, \\ \lceil \frac{n_0}{2} \rceil + \left\lfloor \frac{m}{2m-1}n_{\mathsf{oth}} - \frac{m-1}{2(2m-1)}n_0 \right\rfloor, & \text{otherwise.} \end{cases} \tag{5}$$

Proof. The correctness of the protocol is straightforward. The CSP-fairness of the protocol follows by combining Lemma 4.4, Lemma 4.6, and Lemma 4.7, where t can be obtained by substituting $n_{\mathsf{oth}} = n - n_0$ in Table 1. □

Landscape visualization. One can check that as m grows, the size t of the coalition Protocol 4.3 can tolerate gradually decreases. In **??**, we give an illustration of t as a function of n_0 and n_{oth} for $m = 2, 3, 4$ in **??**. This helps to understand the mathematical landscape of CSP-fairness.

5 Non-Uniform Coin-Toss: Semi-Malicious Security

In this section, we give a fair coin-toss protocol for a non-uniform coin. Suppose that we want to toss an m-coin $\mathcal{D}$ defined by a distribution $\{p_0, \ldots, p_{m-1}\}$ on $\{0, \ldots, m-1\}$. Without loss of generality, we assume that $0 < p_j < 1$ for all $j = 0, \ldots, m-1$ (otherwise, it reduces to a coin with fewer sides).

To ensure the correctness of the coin-toss, the players will toss δ (the correctness parameter) number of binary coins *in parallel* and map the δ-bits string to an m-coin. The mapping is straightforward: We partition the integers $[0, 2^\delta - 1]$ into m parts according to the distribution $\mathcal{D}$. Formally, let $\mathsf{map}_{\mathcal{D}}$ be a function, parametrized with the distribution $\mathcal{D}$, that takes in a δ-bits long binary string $\boldsymbol{\tau}$ and outputs a number $s \in \{0, \ldots, m-1\}$. Formally, $\mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau}) = s$ iff $2^\delta \cdot \sum_{j=0}^{s-1} p_j \le \mathsf{int}(\boldsymbol{\tau}) < 2^\delta \cdot \sum_{j=0}^{s} p_j$, where $\mathsf{int}(\boldsymbol{\tau})$ denotes the integer represented by the binary string $\boldsymbol{\tau}$, and we use $\sum_{j=0}^{-1} p_j = 0$ for simplicity. It is easy to verify that if $\boldsymbol{\tau}$ is uniformly randomly sampled from $\{0,1\}^\delta$, then the distribution of $\mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau})$ is negligibly close to $\{p_0, \ldots, p_{m-1}\}$. Equivalently, for any $j = 0, \ldots, m-1$, we have $\left|\Pr\left[\boldsymbol{\tau} \xleftarrow{\$} \{0,1\}^\delta : \mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau}) = j\right] - p_j\right| \le \mathsf{negl}(\delta)$.

To guarantee game-theoretic fairness against a coalition of size t, the key idea is the same as in the uniform coin-toss: we want to incentivize the successful reconstruction of the last coin. If the reconstruction of the last coin fails, the protocol outputs the "least preferred outcome", such that no coalition of size no more than t is incentivized to fail the reconstruction. However, the "least preferred outcome" for a non-uniform coin-toss is not simply determined by the number of supporters for each outcome. Instead, it is determined by the preference profile as well as the desired distribution. For each $j = 0, \ldots, m-1$, define the balancing parameter p_j^* based on $\mathcal{D}$ as follows:

$$p_j^* = \frac{\min_{i \neq j}(p_i)}{1 - p_j}. \tag{6}$$

Roughly speaking, this p_j^* measures the maximum proportion of the number of j-supporters versus the number of other supporters in the coalition, such that the coalition does NOT have incentives to bias the outcome j. The least preferred outcome lpo is thus defined as $\mathsf{lpo} := \arg\max_j \left(p_j^*(n - n_j) - n_j\right)$. For simplicity, in this section, we assume that 0 is the least preferred outcome, and we use p^* to denote p^*_{lpo}.

Based on the mapping function $\mathsf{map}_{\mathcal{D}}$ and the definition of the least preferred outcome, we can now proceed to describe the protocol for tossing arbitrary m-coins. Still, the coin-toss protocol runs two instances of the $\mathsf{GroupToss}$ sub-protocol, $\mathsf{GroupToss}^0[2, k_0]$ among the group of 0-supporters (because it is the least preferred outcome), and
$\mathsf{GroupToss}^{\mathsf{oth}}[2, k_{\mathsf{oth}}]$ among the others.

Protocol 5.1 (Fair m-coin-toss with semi-malicious security).
Input: the target distribution $\mathcal{D} = \{p_0, \ldots, p_{m-1}\}$, and the preference profile $[n_0, \ldots, n_{m-1}]$. Let 0 be the least preferred outcome.
Sharing phase

- 0-supporters run δ independent, parallel instances of the sharing phase of $\mathsf{GroupToss}^0[2, k_0]$. Other players observe.
- Other supporters together run δ independent, parallel instances of the sharing phase of $\mathsf{GroupToss}^{\mathsf{oth}}[2, k_{\mathsf{oth}}]$. The 0-supporters observe.

Reconstruction phase.

- 0-supporters run the reconstruction phase of the δ parallel instances of $\mathsf{GroupToss}^0[2, k_0]$. Other players observe. If the reconstruction in all sessions succeeds, let

s_0^{sid} be the outcome of session sid. Define the binary string $\boldsymbol{\tau}_0 = (s_0^1, \ldots, s_0^\delta)$. Otherwise, if there exists a session that fails to reconstruct, let $\boldsymbol{\tau}_0 = (0, \ldots, 0)$ of length δ.

- Other supporters run the reconstruction phase of the δ instances of $\mathsf{GroupToss}^{\mathsf{oth}}[2, k_{\mathsf{oth}}]$. The 0-supporters observe. If there exists a session in which the reconstruction fails, output $s = 0$ as the *final coin value*. Otherwise, let s_{oth}^{sid} be the outcome of session sid. Define the binary string $\boldsymbol{\tau}_1 = (s_{\mathsf{oth}}^1, \ldots, s_{\mathsf{oth}}^\delta)$. The protocol outputs $s = \mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau}_0 \oplus \boldsymbol{\tau}_{\mathsf{oth}})$.

Note that all the properties of GroupToss still hold for the δ parallel instances.

Threshold parameter selection. Suppose the coalition controls $t = t_0 + t_{\mathsf{oth}}$ number of players. To achieve CSP-fairness, we need to guarantee (C1) and (C2), as in the uniform coss scenario. The only difference is about the third condition: When we say the coalition must NOT prefer 0 compared to the honest behavior, the honest expected utility is now $\sum p_j t_j$. Formally,

(C3′) *If the coalition can fail the reconstruction of $\boldsymbol{\tau}_{\mathsf{oth}}$, it must NOT prefer* 0. If the coalition controls at least $n_{\mathsf{oth}} - k_{\mathsf{oth}}$ number of other supporters such that it can choose to fail the reconstruction of s_{oth}^{sid} for any $sid \in [\delta]$ and force the protocol to output 0, then its joint utility must not increase. In other words, if $t_{\mathsf{oth}} \geq n_{\mathsf{oth}} - k_{\mathsf{oth}}$, then the number of corrupted 0-supporters must satisfy $t_0 \leq \sum_{j=0}^{m-1} p_j t_j$, where t_j is the number of j-supporters in the coalition.

If conditions (C1), (C2) and (C3′) are all satisfied, then Protocol 5.1 satisfies CSP-fairness against any t-sized coalition.

Lemma 5.2. *Suppose that the parameters k_0 and k_{oth} are chosen such that conditions (C1), (C2), (C3′) are all satisfied, then Protocol 5.1 satisfies CSP-fairness against any semi-malicious coalition of size no more than t.*

Proof. By the same proof as Lemma 4.4. Due to condition (C3′), it never makes sense for the coalition to fail the reconstruction of s_{oth}^{sid} for any single instance $sid \in [\delta]$. Then by (C1) and (C2), the binary string $\boldsymbol{\tau}_0 \oplus \boldsymbol{\tau}_{\mathsf{oth}}$ must be a uniformly random string. Therefore, the coalition cannot increase its joint utility. □

Now we only need to care about choosing the parameters k_0 and k_{oth} such that conditions (C1), (C2) and (C3′) are satisfied. Below, we give the parameter constraints for the non-uniform coin-toss. Recall that $p^* = \frac{\min(p_1, \ldots, p_{m-1})}{1 - p_0}$.

Parameter Constraints 5.3 (Arbitrary m-coin, semi-malicious version)**.**

- $0 \leq k_0 \leq n_0$, $0 \leq k_{\mathsf{oth}} \leq n_{\mathsf{oth}}$
- $t \leq k_0 + k_{\mathsf{oth}} + 1$,
- $t \leq n_0 - k_0 + k_{\mathsf{oth}}$,
- If $n_{\mathsf{oth}} - k_{\mathsf{oth}} < \frac{1}{p^*} n_0$, then $t \leq (1 + p^*)(n_{\mathsf{oth}} - k_{\mathsf{oth}})$.

Lemma 5.4. *If k_0 and k_{oth} satisfy Parameter Constraints 5.3, then it satisfies conditions (C1), (C2) and (C3′).*

Proof. We only need to prove that the last constraint implies condition (C3′). Suppose that the coalition is able to fail the reconstruction of $\boldsymbol{\tau}_{\mathsf{oth}}$, i.e., $t_{\mathsf{oth}} \geq n_{\mathsf{oth}} - k_{\mathsf{oth}}$. There are two possible cases:

Case 1: If $n_{\mathsf{oth}} - k_{\mathsf{oth}} \geq \frac{1}{p^*} n_0$. Then $t_{\mathsf{oth}} \geq \frac{1}{p^*} n_0$. Since the number of 0-supporters in the coalition must satisfy $t_0 \leq n_0$, we have $t_0 \leq p^* \cdot t_{\mathsf{oth}}$. This implies $t_0 \leq \sum_{j=0}^{m-1} p_j \cdot t_j$ by definition of p^*. Thus, condition (C3′) is automatically satisfied under this case.

Case 2: If $n_{\mathsf{oth}} - k_{\mathsf{oth}} < \frac{1}{p^*} n_0$, then we have $t \leq (1 + p^*)(n_{\mathsf{oth}} - k_{\mathsf{oth}}) \leq (1 + p^*) t_{\mathsf{oth}}$ by the last constraint in Parameter Constraints 4.5. This implies $t_0 \leq \sum_{j=0}^{m-1} p_j \cdot t_j$ by the same reasoning as in Case 1, we have that condition (C3′) is satisfied. $\square$

Optimal resilience for Protocol 5.1 Similar to the uniform coin-toss, we can now solve for the optimal resilience by finding the maximum t such that there exists a feasible solution for k_0 and k_{oth} satisfying Parameter Constraints 5.3. Below we give the optimal resilience for a m-way coin $\mathcal{D} = \{p_0, \ldots, p_{m-1}\}$ given a preference profile $[n_0, \ldots, n_{m-1}]$. We omit the formal proof since it is similar to that of the uniform coin case.

Case	k_0	k_{oth}	t
If $n_{\mathsf{oth}} \geq \frac{p^*+4}{2p^*} n_0$	$\lfloor \frac{n_0}{2} \rfloor$	$\lfloor n_{\mathsf{oth}} - \frac{1}{p^*} n_0 \rfloor$	$\lceil \frac{n_0}{2} \rceil + \lfloor n_{\mathsf{oth}} - \frac{1}{p^*} n_0 \rfloor$
Otherwise	$\lfloor \frac{n_0}{2} \rfloor$	$\left\lfloor \frac{p^*+1}{p^*+2} n_{\mathsf{oth}} - \frac{1}{2(p^*+2)} n_0 \right\rfloor$	$\lceil \frac{n_0}{2} \rceil + \left\lfloor \frac{p^*+1}{p^*+2} n_{\mathsf{oth}} - \frac{1}{2(p^*+2)} n_0 \right\rfloor$

Table 2: Optimal resilience of Protocol 5.1 against semi-malicious adversary.

Notice that as long as $p^* n_{\mathsf{oth}} \geq n_0$ or $n_{\mathsf{oth}} \geq \frac{p^*+4}{2p^*} n_0$, the protocol can tolerate half or larger-sized coalition. Below, we give some interesting examples of the desired distribution of the coin and the size of the coalition we can tolerate.

- For uniform m-coin, it is always the case that $p^* = \frac{1}{m-1}$. Thus, the least preferred outcome is the group with the least number of supporters, and the size of the coalition we can tolerate collapse to the term given in Table 1.

- For arbitrary binary coin, it is always the case that $p^* = 1$. Therefore, the size of the coalition we can tolerate is the same as with the uniform case, which is optimal according to [WAS22].

- For ternary coin $\{\frac{1}{5}, \frac{3}{5}, \frac{1}{5}\}$ and preference profile $n_0 = 4, n_1 = 5, n_2 = 8$, the least preferred outcome is 1, and $p^* = \frac{1}{2}$. In this case, we can tolerate any coalition of size up to 8. One can check that had we used 0 as the least preferred outcome, the resulting protocol only tolerates 6-sized coalition.

Visualizing the landscape In ??, we illustrate the landscape for different ternary coin distributions. For simplicity, we give t as a function of n_0 and n_{oth} for different p^*, assuming 0 is the least preferred outcome. One can check that for fixed n_0 and n_{oth}, the size of the coalition we can tolerate increases as p^* increases. Moreover, the size of the coalition we can tolerate for $p^* = \frac{1}{q}$ for an integer q is actually the same as that for uniform $(\frac{1}{p^*} + 1)$-sided coin.

6 Fair Coin-Toss Protocol for Degenerate Preference Profile: Semi-Malicious Security

So far we have been focused on full preference profiles, i.e., each outcome has at least one supporter. For degenerate preference profiles where there exists an outcome that no player prefers, we can design a protocol that tolerates any coalition of size $n-1$.

Without loss of generality, assume that $n_0 = 0$, i.e., no player prefers outcome 0. In this case, we can simply run a $\mathsf{GroupToss}$ protocol among *all* players. Had the reconstruction failed, the protocol outputs 0 as a punishment for misbehavior. Formally,

Protocol 6.1 (**Fair m coin-toss with semi-malicious security: degenerate preference).**
Input: A *degenerate* preference profile $[n_0, \ldots, n_{m-1}]$ where $n_0 = 0$.

Sharing phase: all players run δ independent, parallel instances of the sharing phase of $\mathsf{GroupToss}[2, n-1]$.

Reconstruction phase: all players run the reconstruction phase of the δ parallel instances of $\mathsf{GroupToss}[2, n-1]$. If any single instance fails, output 0; otherwise, let $\boldsymbol{\tau}$ be the reconstructed value. Output $\mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau})$.

Theorem 6.2. *Protocol 6.1 is a CSP-fair coin-toss protocol for degenerate preference profiles against any semi-malicious coalition of size no more than $n-1$.*

Proof. By the knowledge threshold property of $\mathsf{GroupToss}$, the coalition's view at the end of the sharing phase is independent of the coin value s that the sharing phase binds to. If the reconstruction fails, the coalition gets utility 0 since no one prefers the outcome 0. Therefore, it never makes sense for the coalition to fail the reconstruction. If the reconstruction succeeds, the output is a uniformly random coin by the knowledge threshold property. □

7 Upgrade to Malicious Security

So far, we have been focused on getting game-theoretic fairness against semi-malicious coalitions. In this section, we will show how to upgrade the protocol to be fair against even malicious coalitions. Again, we will focus on the non-degenerate preference profiles and give the corresponding protocols for degenerate preference profiles with malicious security at the end of this section. The upgrade is based on multi-party computation with identifiable abort, signature scheme, and commitment scheme. We first give a detailed description of the functionality that these building blocks provide.

7.1 Building Blocks for Protocol 7.2

In this section, we introduce the building blocks used in Protocol 7.2.

Identificable abort Roughly speaking, a multi-party computation with identifiable abort works as follows: either the players securely compute some function $\mathcal{F}$, or if the computation fails, honest players receive the identity of misbehaved players. In this part, we only give the ideal-world execution of computing $\mathcal{F}$ with identifiable abort, assuming the existence of a trusted authority. Previous work [MGW87, Gol04, IOZ14] showed how to instantiate this trusted authority, and we refer the readers to [IOZ14] for more details.

Let $\mathcal{A}$ be an adversary that controls a set of parties $I \subset [n]$. The adversary $\mathcal{A}$ gets corrupted players' input and controls the messages it sends. Given a function $\mathcal{F}$, the ideal execution works as follows:

- Each player i gets the security parameter λ. The adversary $\mathcal{A}$ may receive auxiliary inputs.
- The trusted authority uniformly samples some randomness r and computes the output $(y_1, \ldots, y_n)$ of the function $\mathcal{F}$[3], based on the security parameter λ and r. The trusted authority then sends $\{y_i\}_{i \in I}$ to $\mathcal{A}$.
- Upon receiving the outputs, the adversary $\mathcal{A}$ either sends ok to the trusted authority, or abort_i for corrupted player i.
- If the adversary sends ok, the trusted authority sends the honest output y_j to honest player $j \notin I$. Otherwise, it sends abort_i to each honest player.
- The honest players output whatever they received from the trusted authority, the corrupted players output nothing, and the adversary $\mathcal{A}$ outputs an arbitrary function based on its view.

The following lemma comes from [MGW87, Gol04, IOZ14].

Lemma 7.1. *Assuming the existence of oblivious transfers. For any n-party functionality with no inputs, there exists a protocol Π that securely computes $\mathbb{F}$ with identifiable abort.*

Signature scheme A signature scheme Sig consists of three algorithms:

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$: The key generation algorithm KeyGen takes as input a security parameter λ and outputs a signing key sk and a verification key vk.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$: the signing algorithm takes in the signing key key and a message msg, and outputs a signature σ.
- $b \leftarrow \mathsf{Verif}(\mathsf{vk}, \mathsf{msg}, \sigma)$: the verification key takes as input the verification key vk, the message msg, the signature σ, and outputs 1 if the signature is valid, and 0 otherwise.

A signature scheme must satisfy the following two properties:

- *Correctness*: For any messages msg, the following holds

$$\Pr[\mathsf{Verif}(\mathsf{vk}, \mathsf{msg}, \sigma) = 1 : (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda), \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})] = 1.$$

- *Security*: For any non-uniform p.p.t. adversary $\mathcal{A}$, after querying signatures for polynomially many messages $\{\mathsf{msg}_i\}_{1 \le i \le \mathsf{poly}(\lambda)}$, the chance that $\mathcal{A}$ can forge a valid signature for a message msg' that does not belong to $\{\mathsf{msg}_i\}_{1 \le i \le \mathsf{poly}(\lambda)}$ is negligible.

Commitment scheme A commitment scheme is separated into two phases:

- **Commit phase**: Given a message msg the committer sends $\mathsf{com} = \mathsf{Commit}(\mathsf{msg}, \mathsf{r})$ for some randomness r to the receiver.
- **Decommit phase**: The committer sends the pair (msg, r) to the receiver. The receiver verifies if the pair is indeed correct by computing $\mathsf{Commit}(\mathsf{msg}, \mathsf{r})$, and compares it with the previously received com. If they are equal, the receiver accepts. Otherwise, the receiver rejects.

[3] While we focus on inputless functions, the results does hold for functions with inputs.

A commitment scheme must satisfy the following properties.

- *Binding*: For any $\mathsf{msg}_0 \neq \mathsf{msg}_1$, for any r_0, r_1,

$$\mathsf{Commit}(\mathsf{msg}_0, \mathsf{r}_0) \neq \mathsf{Commit}(\mathsf{msg}_1, \mathsf{r}_1).$$

- *Hiding*: For any msg_0 and msg_1,

$$\{\mathsf{Commit}(\mathsf{msg}_0, \mathsf{r}) : r \overset{\$}{\leftarrow} U\} \equiv_c \{\mathsf{Commit}(\mathsf{msg}_1, \mathsf{r}), r \overset{\$}{\leftarrow} U\}.$$

7.2 Malicious Secure GroupToss Sub-Protocol

We adopt the same upgrade as in [WAS22] to get malicious security for GroupToss. Specifically, they build a counterpart of the sub-protocol GroupToss using MPC with identifiable abort [IOZ14], which can be realized assuming the existence of oblivious transfer (OT). In MPC with identifiable abort, either the players securely compute some functions, or if the computation fails, honest players receive the identity of misbehaved players. In this case, the honest players can kick out the offending players and redo the computation till they succeed. For completeness, we present the full protocol $\mathsf{GroupToss}^b[\ell, k; \theta]$ with malicious security below. Roughly speaking, $\mathsf{GroupToss}^b[\ell, k; \theta]$ outputs a string in the space of $\{0, \dots, \ell - 1\}^\theta$. Still, the parameter k denotes the threshold parameter.

Protocol 7.2 ($\mathsf{GroupToss}^b[\ell, k; \theta]$ **sub-protocol (malicious security)).**
Sharing Phase

1. Let the active set $\mathcal{O} := \mathcal{P}_b$. Repeat the following until success:
   (a) The active set $\mathcal{O}$ securely computes the ideal functionality $\mathcal{F}^{b,\mathcal{O}}_{\text{sharegen}}[\ell, k; \theta]$ to be described below (Functionality 7.3) using MPC with identifiable abort.
   (b) If the computation aborts, then every honest player obtains the identity of a misbehaved player j^*. Remove j^* from $\mathcal{O}$.
2. At this moment, each player $i \in \mathcal{O}$ has obtained a tuple $(\mathsf{vk}, [s]_i, [r]_i, [\mathsf{com}]_i, \sigma_i, \sigma'_i)$ from $\mathcal{F}^{b,\mathcal{O}}_{\text{sharegen}}[\ell, k; \theta]$.

Voting phase.

1. Each player posts vk as its vote to the broadcast channel. Let vk' be the verification key that has gained the most votes, breaking ties arbitrarily.
2. If vk' has gained at most k votes, the voting phase fails.

   Else, player i posts $[\mathsf{com}]_i$ and σ_i to the broadcast channel if vk' is the same as the vk it received from $\mathcal{F}^{b,\mathcal{O}}_{\text{sharegen}}[\ell, k; \theta]$. Otherwise, do nothing.
3. Everyone gathers all $([\mathsf{com}]_j, \sigma_j)$ pairs posted to the broadcast channel such that σ_j is a valid signature of $[\mathsf{com}]_j$ under vk'. If there are at least $k + 1$ such tuples and all shares $[\mathsf{com}]_j$ reconstruct uniquely to the value com, then record the reconstructed commitment com. Else we say that the vote phase failed.

Reconstruction phase.

1. If the vote phase failed, output the reconstructed value $\perp$. Else, continue with the following.

2. For each player $i \in \mathcal{O}$, if $\mathsf{vk}' = \mathsf{vk}$, then post to the broadcast channel the tuple $([s]_i, [r]_i, \sigma'_i)$.
3. Every player does the following: gather all tuples $([s]_j, [r]_j, \sigma'_j)$ posted to the broadcast channel such that σ'_j is a valid signature for $([s]_j, [r]_j)$ under vk'. If all such $([s]_j, [r]_j)$ tuples reconstruct to a unique value $(\mathbf{s}, \mathbf{r})$ and moreover, $(\mathbf{s}, \mathbf{r})$ is a valid opening of com, then output the reconstructed value $\mathbf{s}$. Else, output $\perp$ as the reconstructed value.

Functionality 7.3 **(The $\mathcal{F}^{b,\mathcal{O}}_{\text{sharegen}}[\ell, k; \theta]$ ideal functionality).**

1. Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$ where $\mathsf{Sig} := (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verif})$ denotes a signature scheme.
2. Sample $\mathbf{s} \overset{\$}{\leftarrow} \{0, \ldots, \ell - 1\}^\theta$, and randomness $\mathbf{r} \in \{0, 1\}^\lambda$, let $\mathsf{com} := \mathsf{Commit}(\mathbf{s}, \mathbf{r})$.
3. Use a $(k+1)$-out-of-$|\mathcal{O}|$ Shamir secret sharing scheme to split the terms $(\mathbf{s}, \mathbf{r})$ and com into $|\mathcal{O}|$ shares, denoted $\{[s]_i, [r]_i, [\mathsf{com}]_i\}_{i \in \mathcal{O}}$, respectively. Let $\sigma_i := \mathsf{Sig.Sign}(\mathsf{sk}, [\mathsf{com}]_i)$ and $\sigma'_i := \mathsf{Sig.Sign}(\mathsf{sk}, ([s]_i, [r]_i))$ for $i \in \mathcal{O}$.
4. Each player in $\mathcal{O}$ receives the output $(\mathsf{vk}, [s]_i, [r]_i, [\mathsf{com}]_i, \sigma_i, \sigma'_i)$.

The above Protocol 7.2 satisfies the following properties:

- *Binding.* If the voting phase does not fail, then the broadcast messages in the sharing and voting phase uniquely bind to a secret s, such that the reconstruction phase either output s or $\perp$.
- *Knowledge threshold.* If at least $k+1$ number of b-supporters are in the coalition, then they can bias coin values s that the sharing and vote phases uniquely bind to, had the voting phase succeeded.

  On the other hand, if the coalition controls at most k number of b-supporters, then the coalition's view at the end of the voting phase is *computationally independent* of the coin value s that the sharing and voting phase bind to. Formally, for any non-uniform p.p.t. coalition controlling at most k number of b-supporters, there exists a there exists a simulator $\mathcal{S}(1^\lambda)$ such that either the voting phase fails, or
  $$(s, \mathsf{view}_\mathcal{A}) \equiv_c (\mathsf{Uniform}, \mathcal{S}(1^\lambda)), \tag{7}$$
  where s denotes the unique coin that the sharing phase and voting phase bind to, $\mathsf{view}_\mathcal{A}$ denotes the coalition's view at the end of the voting phase, and $\mathsf{Uniform}$ denotes a value uniformly randomly sampled from $\{0, \ldots, m-1\}$.
- *Liveness threshold.* If the coalition controls at least $\min(n_b - k, \lceil n_b/2 \rceil)$ number of b-supporters, it can cause the reconstruction to output $\perp$. On the other hand, if the coalition controls fewer than $\min(n_b - k, \lceil n_b/2 \rceil)$ number of b-supporters, then the reconstruction phase must succeed.

Lemma 7.4. *Assuming the existence of OT, Protocol 7.2 satisfies binding, knowledge threshold, and liveness threshold properties.*

Proof. The proof is similar to the proof of Lemma C.1 in [WAS22], except that the only difference is that now the ideal functionality $\mathcal{F}_{\text{sharegen}}$ shares $\mathbf{s}$ from $\{0, \ldots, \ell - 1\}^\theta$ instead of a single binary coin. Despite these changes, the security proof is still the same. □

7.3 Fair Coin-Toss Protocol: Malicious Security

Now, we proceed to build the CSP-fair protocol against malicious coalitions based on the malicious version of $\mathsf{GroupToss}$. Without loss of generality, assume that 0 is the least preferred outcome, and

p^* is the corresponding balancing parameter.

Protocol 7.5 (Fair m-coin-toss with malicious security).
Input: The target distribution $\mathcal{D} = \{p_0, \ldots, p_{m-1}\}$, a non-degenerate preference profile $[n_0, \ldots, n_{m-1}]$, where 0 is the least preferred outcome.
Parameter Selection: For uniform coin-toss, set $\ell = m$ and $\theta = 1$; for non-uniform coin-toss, set $\ell = 2$ and $\theta = \delta$, where δ is the correctness parameter. We will specify how to choose k_0 and k_{oth} later.

Sharing phase

(S-0) 0-supporters run the sharing phase of $\mathsf{GroupToss}^0[\ell, k_0; \theta]$. Other players observe.

(S-oth) Other supporters run the sharing phase of $\mathsf{GroupToss}^{\mathsf{oth}}[\ell, k_{\mathsf{oth}}; \theta]$. The 0-supporters observe.

Voting phase (The order here is important).

(V-oth) Other supporters run the voting phase of $\mathsf{GroupToss}^{\mathsf{oth}}[\ell, k_{\mathsf{oth}}; \theta]$. The 0-supporters observe.

(V-0) 0-supporters run the voting phase of $\mathsf{GroupToss}^0[\ell, k_0; \theta]$. Other players observe.

Reconstruction phase.

(R-0) 0-supporters run the reconstruction phase of $\mathsf{GroupToss}^0[\ell, k_0; \theta]$. Other players observe. If the reconstruction succeeded, let its outcome be $\boldsymbol{\tau}_0$. Otherwise, set $\boldsymbol{\tau}_0 = 0^\theta$.

(R-oth) Other supporters run the reconstruction phase of $\mathsf{GroupToss}^{\mathsf{oth}}[\ell, k_{\mathsf{oth}}; \theta]$. The 0-supporters observe. If the reconstruction fails, output $s = 0$ as the *final coin value.* If the reconstruction succeeds, let its outcome be $\boldsymbol{\tau}_{\mathsf{oth}}$.

For uniform coin-toss, output $(\boldsymbol{\tau}_0 + \boldsymbol{\tau}_{\mathsf{oth}}) \mod m$ as the final coin value. Otherwise, output $\mathsf{map}_{\mathcal{D}}(\boldsymbol{\tau}_0 \oplus \boldsymbol{\tau}_{\mathsf{oth}})$.

Threshold parameter selection. Suppose the coalition controls $t = t_0 + t_{\mathsf{oth}}$ number of players, where t_0 and t_{oth} are the number of 0-supporters and other supporters in the coalition, respectively. Now we can write down a counterpart of conditions (C1), (C2), and (C3) with malicious security:

(C1*) *The coalition cannot control both coins.* That is, if the coalition controls at least k_0+1 number of 0-supporters, then the coalition must control at most k_{oth} number of other supporters due to the corruption budget t, and vise versa.

(C2*) *If the coalition controls the s_{oth} coin, it cannot hamper the reconstruction of the coin s_0.* If the coalition controls at least $k_{\mathsf{oth}} + 1$ number of other supporters, then due to its corruption budget, it must control less than $\min(n_0 - k_0, \lceil n_0/2 \rceil)$ number of 0-supporters.

(C3*) *If the coalition can fail the reconstruction of the s_{oth} coin, it must NOT prefer* 0. If the coalition controls at least $\min(n_{\mathsf{oth}} - k_{\mathsf{oth}}, \lceil n_{\mathsf{oth}}/2 \rceil)$ number of other supporters such that it can choose to fail the reconstruction of s_{oth} and force the protocol to output 0, then its joint utility must not increase. In other words, if $t_{\mathsf{oth}} \geq \min(n_{\mathsf{oth}} - k_{\mathsf{oth}}, \lceil n_{\mathsf{oth}}/2 \rceil)$, then the number of corrupted 0-supporters must satisfy $t_0 \leq \sum p_j t_j$.

Given a preference profile $[n_0, \ldots, n_{m-1}]$, and the corruption budget t, if there exists a feasible solution for k_0 and k_{oth} such that the above three constraints are all satisfied, then Protocol 7.5

parametrized with the feasible solution k_0 and k_{oth} satisfies CSP-fairness against any t-sized malicious coalition.

Lemma 7.6. *Assuming the existence of OT. If the parameters k_0 and k_{oth} are chosen such that conditions (C1*), (C2*), (C3*) are all satisfied, then Protocol 7.5 satisfies CSP-fairness against any non-uniform p.p.t. malicious coalition of size no more than t.*

Proof. First, due to condition (C3*), it never makes sense for the coalition to fail the reconstruction of the s_{oth}. Henceforth we only need to prove that for any non-uniform p.p.t. coalition such that s_{oth} is successfully reconstructed, the outcome $s_0 + s_{\mathsf{oth}}$ is computationally indistinguishable from a uniform random coin. There are two cases.

Case 1. If $t_{\mathsf{oth}} \leq k_{\mathsf{oth}}$: Consider the following sequence of hybrids, where $\mathcal{A}$ denotes the non-uniform p.p.t. adversary that controls the coalition.

- Real: A real execution of GroupToss from Step (S-0) to (R-0) in the $\mathcal{F}_{\text{sharegen}}$-hybrid world. Observe that at this moment, $s_0 + s_{\mathsf{oth}}$ is well-defined.
- Hybrid: A simulator $\mathcal{S}^*$ interacts with the adversary $\mathcal{A}$ as follows:
  - $\mathcal{S}^*$ acts on behalf of the honest players and interacts with $\mathcal{A}$ in Step (S-0). Let state denote the adversary's current state.
  - Then, in Step (S-oth) and (V-oth), the simulator sample $s_{\mathsf{oth}} \xleftarrow{\$} \{0, \ldots, \ell - 1\}^{\theta}$, runs $\mathcal{S}(1^{\lambda})$ defined in (7) given the adversary's state state, and reset $\mathcal{A}$'s state based on the output of the simulator.
  - Then $\mathcal{S}^*$ continues to act as the honest players and interact with $\mathcal{A}$. Still, s_0 and s_{oth} are well-defined.

  By the construction, $s_0 + s_{\mathsf{oth}}$ in Hybrid is uniformly random. By the knowledge threshold property, when $t_{\mathsf{oth}} \leq k_{\mathsf{oth}}$, these two hybrids Real and Hybrid are indistinguishable. Therefore, $s_0 + s_{\mathsf{oth}}$ in Real is computationally indistinguishable from a uniformly random coin.

Case 2. If $t_{\mathsf{oth}} > k_{\mathsf{oth}}$: By Condition (C1*) and (C2*), it must be that $t_0 \leq k_0$, and s_0 must be successfully reconstructed. This implies that at the end of Step (V-oth), the coins s_0 and s_{oth} are determined. Therefore, we can use a similar proof as in Case 1. Formally, consider the following sequence of hybrids, where $\mathcal{A}$ denotes the non-uniform p.p.t. adversary that controls the coalition.

- Real: A real execution of GroupToss from Step (S-0) to (V-0) in the $\mathcal{F}_{\text{sharegen}}$-hybrid world. Observe that at this moment, $s_0 + s_{\mathsf{oth}}$ is well-defined.
- Hybrid: A simulator $\mathcal{S}^*$ acts on behalf of the honest players and interacts with the adversary $\mathcal{A}$ as follows: it samples $s_0 \xleftarrow{\$} \{0, \ldots, \ell - 1\}^{\theta}$, runs $\mathcal{S}(1^{\lambda})$ defined in (7) from Step (S-0) to (V-0). At this point, $s_0 + s_{\mathsf{oth}}$ is well-defined.

  By the construction, $s_0 + s_{\mathsf{oth}}$ in Hybrid is uniformly random. By the knowledge threshold property, when $t_0 \leq k_0$, these two hybrids Real and Hybrid are indistinguishable. Therefore, $s_0 + s_{\mathsf{oth}}$ in Real is computationally indistinguishable from a uniformly random coin.

□

Now the problem boils down to solving the optimization problem that maximizes t subject to the above three conditions. One can easily verify that any k_0, k_{oth}, t that satisfy (C1*), (C2*), (C3*) must also satisfy the earlier conditions (C1), (C2) and (C3). This implies that the malicious version of the protocol cannot tolerate a larger-sized coalition than the semi-malicious version. Interestingly, it turns out that for uniform coins, the choice of k_0 and k_{oth} that maximizes t for the semi-malicious version in Table 1 satisfy (C1*), (C2*), (C3*).

Lemma 7.7. *Assuming the existence of OT. Suppose $p^* \cdot n_{\mathsf{oth}} \geq n_0$. Let k_0 and k_{oth} be chosen as in Table 2. Then conditions (C1*), (C2*), (C3*) are all satisfied.*

Proof. Condition (C1*) and (C2*) are straightforward. We only prove this for (C3*). Since $\frac{p^*+1}{p^*+2}n_{\mathsf{oth}} - \frac{1}{2(p^*+2)}n_0 \geq \frac{1}{2}n_{\mathsf{oth}}$ by the assumption that $p^* \cdot n_{\mathsf{oth}} \geq n_0$, we have $k_{\mathsf{oth}} \geq \lfloor \frac{1}{2}n_{\mathsf{oth}} \rfloor$. Thus, $\min(n_{\mathsf{oth}} - k_{\mathsf{oth}}, \lceil n_{\mathsf{oth}}/2 \rceil) = n_{\mathsf{oth}} - k_{\mathsf{oth}}$. Therefore, by the same reasoning as in Lemma 4.6, condition (C3*) is satisfied under the given parameters. □

Combining the above results, we have the following theorem.

Theorem 7.8. *Given a non-degenerate preference profile $[n_0, \ldots, n_{m-1}]$ and the desired m-coin distribution $\mathcal{D}$. Without loss of generality, assume 0 is the least preferred outcome and p^* denote the corresponding balancing parameters (6).*

If $p^* \cdot n_{\mathsf{oth}} \geq n_0$, then Protocol 7.5 is a CSP-fair coin-toss protocol against any t-sized malicious adversary, where t is the same as the one given in Table 2.

Proof. Combining Lemma 7.7, Lemma 7.6, and Lemma 7.4. □

7.4 Fair Coin-Toss Protocol for Degenerate Preference Profile: Malicious Security

The protocol for degenerate preference profiles follows from standard cryptographic techniques of public verifiable concurrent non-malleable commitments [LPV08] and public verifiable concurrent zero-knowledge proofs [Pas04]. Specifically, each player i now shares a random secret c_i as follows: they first post a commitment of c_i and all the shares $[c_i]_j$ in the broadcast channel, attached with a zero-knowledge proof that these shares and commitments are generated honestly. Every player then checks if the proof is valid, and if the shares they received are valid openings of the commitments. If not, they can post a complaint on the broadcast channel. If there is no complaint, the players use their shares to reconstruct the secret. Had anyone misbehaved in the protocol such that there is a complaint, the protocol outputs an outcome that no one prefers as the punishment. Since this is a standard cryptographic technique of upgrading to malicious security (see [KMSW22]), we omit the redundant proof here.

8 Impossibility Results

Our impossibility results extend that of [CGL+18a,WAS22]. The key is to show that if a uniform m-coin-toss protocol $\widetilde{\Pi}$ is CSP-fair for a preference profile $[n_0, \ldots, n_{m-1}]$ against any t-sized coalition, then $\widetilde{\Pi}$ can be transferred into a three-party protocol Π among $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that certain conditions. Below, we first list the set of all conditions.

(LB-C1) **Lone-wolf condition**: a semi-malicious, non-uniform p.p.t. adversary controlling $\mathcal{S}_1$ (or $\mathcal{S}_3$) cannot bias the output towards *any* direction.

(LB-C2) **b-wolf-minion condition**: a semi-malicious non-uniform p.p.t. adversary controlling $\mathcal{S}_1$ and $\mathcal{S}_2$ (or $\mathcal{S}_3$ and $\mathcal{S}_2$) cannot increase the probability of outputs greater than or equal to b.

(LB-C3) T_2**-equity condition**: for all but a negligible fraction of $\mathcal{S}_2$'s randomness T_2, the distribution of $f(T_2)$ is indistinguishable from the uniform distribution, where $f(T_2)$ denotes the outcome in an honest execution where $\mathcal{S}_2$'s randomness is fixed to T_2.

We will show that the three-party protocol must satisfy the lone-wolf condition, wolf-minion condition, and T_2-equity condition. However, the following theorem implies that such three-party protocols do not exist.

Theorem 8.1. *No m-coin-toss protocol Π among three parties $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that terminates in polynomially many rounds can simultaneously satisfy the lone-wolf condition (LB-C1), the b-wolf-minion condition (LB-C2), and the T_2-equity condition (LB-C3) for any $b \in \{1, \ldots, m-1\}$.*

We defer the proof of Theorem 8.1 to Section 8.2. For now, assume that Theorem 8.1 holds, and we now show how to transfer a CSP-fair protocol Π against any t-sized coalition to a three-party protocol that satisfies all three conditions. Without loss of generality, assume that identities are ordered. We first partition n players into three supernodes $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$, each supernodes runs the code of the players it contains.

- $\mathcal{S}_1$ contains the first α_0 number of 0-supporters, first α_1 number of 1-supporters, ..., and the first α_{m-1} number of $(m-1)$-supporters.
- $\mathcal{S}_2$ contains the next $n_0 - 2\alpha_0$ number of 0-supporters, $n_1 - 2\alpha_1$ number of 1-supporters, ..., and the first $n_{m-1} - 2\alpha_{m-1}$ number of $(m-1)$-supporters.
- $\mathcal{S}_3$ contains the remaining players.

By the definition, $\mathcal{S}_1$ and $\mathcal{S}_3$ contains $\sum_{j=0}^{m-1} \alpha_j$ of players, and $\mathcal{S}_2$ contains the rest $n - 2\sum_{j=0}^{m-1} \alpha_j$ number of players. The three-party protocol Π is then run among $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{S}_3$. Without loss of generality, assume that $\mathcal{S}_1$ sends a message in the first round. In the following rounds, every supernode sends messages. Then, in the last round, only $\mathcal{S}_3$ sends messages. Each supernode $\mathcal{S}_i$ internally emulates the execution of all parties it runs and internally deals with all the messages sent between these parties. Messages from a party controlled by $\mathcal{S}_i$ to a party controlled by a different supernode $\mathcal{S}_{i'}$ (here $i \neq i'$) are treated as a message from $\mathcal{S}_i$ to $\mathcal{S}_{i'}$.

8.1 Partitions for Different Preference Profiles

In this section, we show how to partition the n players into three supernodes. Specifically, we will show how to pick $\alpha_0, \ldots, \alpha_{m-1}$. Without loss of generality, we assume that $n_0 \leq n_1 \leq \cdots \leq n_{m-1}$.

Definition 8.2. *We say that a preference profile $[n_0, \ldots, n_{m-1}]$ is b-*bipartite *if there exists feasible solutions for $\alpha_0, \ldots, \alpha_{m-1}$ and $b \in \{1, \ldots, m-1\}$ s.t.*

1. *(Non-negativity): For any $j \in \{0, \ldots, m-1\}$, $0 \leq \alpha_j \leq \min\{n_0 - 1, \lfloor \frac{n_j}{2} \rfloor\}$;*
2. *(Left-partition): $n_0 \geq n_{b-1} - \alpha_{b-1}$.*
3. *(Right-partition): The number of b-supporters $n_b \geq n_{m-1} - \alpha_{m-1} + 1$.*

Below, we prove that for b-bipartite preference profiles, under certain partitions, the resulting three party-protocol must satisfy the lone-wolf condition (LB-C1), b-wolf-minion (LB-C2), and T_2-equity condition (LB-C3).

Lemma 8.3. *Given a b-bipartite preference profile $[n_0, \ldots, n_{m-1}]$ that satisfies Definition 8.2 with $\alpha_0, \ldots, \alpha_{m-1}$. There is no m-coin-toss protocol that is CSP-fair against any t-sized semi-malicious coalition where*

$$t = b \cdot (n_{b-1} - \alpha_{b-1}) + (m - b)(n_{m-1} - \alpha_{m-1} + 1). \tag{8}$$

Proof. We prove that the three-party protocol corresponding to this partition satisfies all three conditions. Since $\mathcal{S}_1$ and $\mathcal{S}_3$ control the same number of players, we only prove the statement for $\mathcal{S}_1$.

Lone-wolf condition. For the sake of contradiction, assume that there exists a strategy S for $\mathcal{S}_1$ that can bias the output towards j^*. Then consider a coalition that contains $\mathcal{S}_1$ and $n_0 - \alpha_{j^*}$ number of j^*-supporters outside $\mathcal{S}_1$, and $n_0 - 1 - \alpha_j$ number of j-supporters outside $\mathcal{S}_1$ for any $j \neq j^*$. According to (Non-negativity), this coalition is well-defined.

This coalition contains n_0 number of j^*-supporters and $n_0 - 1$ number of j-supporters for $j \neq j^*$. By running $\mathcal{S}_1$'s strategy S, this coalition can bias the output towards j^* and get a joint utility of $(n_0 - 1)(1 - p_{j^*}) + n_0 \cdot p_{j^*}$, where p_{j^*} denotes the probability that the protocol outputs j^* when $\mathcal{S}_1$ plays strategy S and all other players behave honestly. This is non-negligibly higher than the honest utility of the coalition, which contradicts CSP-fairness.

Wolf-minion condition. For the sake of contradiction, assume that there exists a semi-malicious strategy S for $\mathcal{S}_1$ and $\mathcal{S}_2$ that increases the probability of getting an output greater than or equal to b. Let p^* denote the probability that the protocol outputs $j \geq b$ when $\mathcal{S}_1$ and $\mathcal{S}_2$ together play the strategy S, and all other players behave honestly. Then $p^* = \frac{m-b}{m} + \epsilon$ for some non-negligible ϵ.

Now consider a coalition that controls $\mathcal{S}_1$, $\mathcal{S}_2$, and $n_{b-1} - \alpha_{b-1} - (n_j - \alpha_j)$ number of j-supporters in $\mathcal{S}_3$ for $j < b$, and $n_{m-1} - \alpha_{m-1} + 1 - (n_j - \alpha_j)$ number of j-supporters in $\mathcal{S}_3$ for $j \geq b$. Then this coalition contains $n_{b-1} - \alpha_{b-1}$ number of j-supporters for $j < b$ and $n_{m-1} - \alpha_{m-1} + 1$ number of j'-supporters for $j' \geq b$. Since the profile is b-bipartite, such a coalition is well-defined. The size of this coalition is $b \cdot (n_{b-1} - \alpha_{b-1}) + (m - b) \cdot (n_{m-1} - \alpha_{m-1} + 1) = t$. In an execution where the players in $\mathcal{S}_1$ and $\mathcal{S}_2$ adopt strategy S and all other players behave honestly, the joint utility of the coalition is $(1 - p^*) \cdot (n_{b-1} - \alpha_{b-1}) + p^* \cdot (n_{m-1} - \alpha_{m-1} + 1)$. Since $p^* = \frac{m-b}{m} + \epsilon$, the coalition gets higher expected utility compared to honest execution. This contradicts CSP-fairness.

T_2-equity. We first show that for all but a negligible fraction of T_2, it must be that $|\Pr[f(T_2) \geq b] - \frac{m-b}{m}| \leq \mathsf{negl}(\lambda)$. By a similar argument as for the wolf-minion condition, one can see that it must be $\Pr[f(T_2) \geq b] \leq \frac{m-b}{m} + \mathsf{negl}(\lambda)$ for any T_2. Otherwise, assume that there exists a randomness T_2^* that this is not true. Consider a coalition $\mathcal{S}^*$ that contains $\mathcal{S}_2$ and $n_b - 2\alpha_b - (n_j - 2\alpha_j)$ number of j-supporters for any $j < b$, and $n_{m-1} - 2\alpha_{m-1} - (n_j - 2\alpha_j)$ number of j'-supporters for any $j' \geq b$. This coalition can increase its expected utility by fixing the randomness to T_2^* and thus increases the probability of getting an output larger than or equal to b, which contradicts CSP-fairness. In addition, since $\mathbf{E}_{T_2}[f(T_2)]$ is negligibly close to the uniform distribution, it must be that $|\Pr[f(T_2) \geq b] - \frac{m-b}{m}| \leq \mathsf{negl}(\lambda)$ for all but a negligible fraction of T_2.

Now suppose that the T_2-equity property is not true, i.e., there exists some randomness that can bias the outcome towards j^* by a non-negligible amount. Without loss of generality, assume that $j^* \geq b$. Now consider a coalition that contains $\mathcal{S}^*$ and another single j^*-supporter outside $\mathcal{S}^*$. This coalition fixes $\mathcal{S}_2$'s randomness to some T_2 that bias the outcome towards j^* and that $|\Pr[f(T_2) \geq b] - \frac{m-b}{m}| \leq \mathsf{negl}(\lambda)$. Let $p^* := \Pr[f(T_2) = j^*]$. Then $p^* = \frac{1}{m} + \epsilon$ for some non-negligible

ϵ. Then, this coalition's expected gain compared to honest utility equals

$$\begin{aligned}
&(n_b - 2\alpha_b)\left(\frac{b}{m} \pm \mathsf{negl}(\lambda)\right) + (n_{m-1} - 2\alpha_{m-1}) \cdot \left(\frac{m-b}{m} \pm \mathsf{negl}(\lambda) - p^*\right) \\
&+ (n_{m-1} - 2\alpha_{m-1} + 1)p^* \\
&- \left((n_b - 2\alpha_b) \cdot \frac{b}{m} + (n_{m-1} - 2\alpha_{m-1}) \cdot \frac{m-b-1}{m} + (n_{m-1} - 2\alpha_{m-1} + 1) \cdot \frac{1}{m}\right) \\
=&\epsilon \pm \mathsf{negl}(\lambda).
\end{aligned}$$

This contradicts CSP-fairness. The lemma thus follows by Theorem 8.1. □

Theorem 8.4. *Given a b-bipartite preference profile $[n_0, \ldots, n_{m-1}]$. There is no CSP-fair m-coin-toss protocol against t-sized semi-malicious coalition, where*

$$t = b \cdot n_0 + (m-b) \cdot \max\left\{n_{m-1} - n_0 + 1, \left\lceil \frac{n_{m-1}}{2} \right\rceil + 1\right\}.$$

Proof. Since the profile is b-bipartite, there exists $\alpha_0, \ldots, \alpha_{m-1}$ such that Definition 8.2 is satisfied. Therefore, one may pick α_{b-1} to be $\min\{n_{b-1} - n_0, \lfloor \frac{n_{b-1}}{2} \rfloor\}$. Moreover, since $n_{m-1} - n_b + 1 \leq \alpha_{m-1} \leq \min\{n_0 - 1, \lceil \frac{n_{m-1}}{2} \rceil\}$, picking $\alpha_{m-1} = \min\{n_0 - 1, \lceil \frac{n_{m-1}}{2} \rceil\}$ always suffices. The theorem thus follows by substituting α_{m-1} into (8). □

Below, we give a few examples to help understand our impossibility results.

- For the binary coin-toss, any preference profile is 1-bipartite. Thus, one can get a more refined impossibility result by substituting suitable α_0 and α_1 into Lemma 8.3 which recovers the impossibility of [WAS22].
- For the ternary coin-toss, if a preference profile is 1-bipartite, and $n_1 + n_2 \geq \frac{9}{2} n_0$, then Theorem 8.4 implies that there is no protocol satisfying CSP-fairness against $2n_2 - n_0 + 1$, while our protocol achieves fairness against any $(n_1 + n_2 - 3n_0/2)$-sized coalition. The gap between our upper bound and the lower bound is no more than $3n_0/2$.
- For any m-sided coin-toss, fully balanced preference profile where all n_i's are equal, it satisfies $(m-1)$-bipartite with $\alpha_j = \lfloor \frac{n_j}{2} \rfloor$ for all values of j. For this case, Lemma 8.3 gives a refined impossibility result: no protocol satisfies CSP-fairness against $m \cdot \lceil \frac{n_0}{2} \rceil + 1$. This implies that our protocol is nearly optimal except for an additive constant loss.

8.2 Proof of Theorem 8.1

Theorem 8.5 (Restatement of Theorem 8.1)**.** *No m-coin-toss protocol Π among three parties $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that terminates in polynomially many rounds can simultaneously satisfy the lone-wolf condition (LB-C1), the b-wolf-minion condition (LB-C2), and the T_2-equity condition (LB-C3) for any $b \in \{1, \ldots, m-1\}$.*

For the sake of contradiction, assume that there exists a protocol Π of R rounds that satisfies all three conditions. The proof of this theorem follows from Cleve's impossibility result. Recall that we assume in protocol Π, the supernode $\mathcal{S}_1$ sends messages in the first round, and then for the following rounds $i = 2, \ldots, R-1$, all three supernodes send messages. In the last round, where $i = R$, only $\mathcal{S}_3$ sends messages.

Consider the following sequence of adversaries for Π as in [Cle86], but now conditioned on that $\mathcal{S}_2$'s randomness is fixed to T_2. Given b that partition the preference profiles, for any $i \in [R]$,

- $\mathcal{A}_i^{<}(1^\lambda, T_2)$ corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$ and tries to bias the output to be smaller than b: $\mathcal{A}_i^b$ fixes $\mathcal{S}_2$'s randomness to T_2, and chooses the randomness for $\mathcal{S}_1$ honestly. It executed the protocol honestly till the moment right before $\mathcal{S}_1$ is going to send its i-th message.

  Then it computes α_i, the output of $\mathcal{S}_1$ and $\mathcal{S}_2$, imagining that $\mathcal{S}_3$ aborts right after sending its message in the i-th round. If $\alpha_i < b$, then $\mathcal{S}_1$ aborts right after sending the i-th message. Otherwise, $\mathcal{S}_1$ aborts before sending the i-th message.

- $\mathcal{A}_i^{\geq}(1^\lambda, T_2)$ corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$ and tries to bias the output towards b or larger: $\mathcal{A}_i^b$ fixes $\mathcal{S}_2$'s randomness to T_2, and chooses the randomness for $\mathcal{S}_1$ honestly. It executed the protocol honestly till the moment right before $\mathcal{S}_1$ is going to send its i-th message.

  Then it computes α_i, the output of $\mathcal{S}_1$ and $\mathcal{S}_2$, imagining that $\mathcal{S}_3$ aborts right after sending its message in the i-th round. If $\alpha_i \geq b$, then $\mathcal{S}_1$ aborts right after sending the i-th message. Otherwise, $\mathcal{S}_1$ aborts before sending the i-th message.

- $\mathcal{B}_i^{<}(1^\lambda, T_2)$ corrupts $\mathcal{S}_3$ and $\mathcal{S}_2$ and tries to bias the output to be smaller than b: $\mathcal{B}_i^b$ fixes $\mathcal{S}_2$'s randomness to T_2, and chooses the randomness for $\mathcal{S}_3$ honestly. It executed the protocol honestly till the moment right before $\mathcal{S}_3$ is going to send its i-th message.

  Then it computes β_i, the output of $\mathcal{S}_3$ and $\mathcal{S}_2$, imagining that $\mathcal{S}_1$ aborts right after sending its message in the i-th round. If $\beta_i < b$, then $\mathcal{S}_3$ aborts right after sending the $(i+1)$-th message. Otherwise, $\mathcal{S}_3$ aborts before sending the i-th message.

- $\mathcal{B}_i^{\geq}(1^\lambda, T_2)$ corrupts $\mathcal{S}_3$ and $\mathcal{S}_2$ and tries to bias the output towards b or larger: $\mathcal{B}_i^b$ fixes $\mathcal{S}_2$'s randomness to T_2, and chooses the randomness for $\mathcal{S}_3$ honestly. It executed the protocol honestly till the moment right before $\mathcal{S}_3$ is going to send its i-th message.

  Then it computes β_i, the output of $\mathcal{S}_3$ and $\mathcal{S}_2$, imagining that $\mathcal{S}_1$ aborts right after sending its message in the i-th round. If $\beta_i \geq b$, then $\mathcal{S}_3$ aborts right after sending the $(i+1)$-th message. Otherwise, $\mathcal{S}_3$ aborts before sending the i-th message.

- $\mathcal{A}_0(1^\lambda, T_2)$ corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$ and wants to bias the output of $\mathcal{S}_3$. It fixes $\mathcal{S}_2$'s randomness T_2 and has $\mathcal{S}_1$ abort at the very beginning of the protocol.

For simplicity, we will ignore the security input 1^λ in the rest of the proof. Given that $\mathcal{S}_2$'s randomness is fixed to T_2, the protocol Π can be viewed as a residual protocol Π_{res} between $\mathcal{S}_1$ and $\mathcal{S}_3$. In Π_{res}, the randomness T_2 is hardwired in $\mathcal{S}_1$ and $\mathcal{S}_3$'s program, so they can simulate the behavior of $\mathcal{S}_2$ on their own. Since the above sequence of adversaries all treat $\mathcal{S}_2$ as a silent corrupted party, they can be transferred into a sequence of adversaries in the residual protocol Π_{res}, where $\{\mathcal{A}_i^{<}(T_2), \mathcal{A}_i^{\geq}(T_2)\}_{i\in[R]}$ and $\mathcal{A}_0(T_2)$ corrupts $\mathcal{S}_1$, and $\{\mathcal{A}_i^{<}(T_2), \mathcal{A}_i^{\geq}(T_2)\}_{i\in[R]}$ corrupts $\mathcal{S}_3$. By a generalization of Cleve [Cle86], we have

Lemma 8.6. *For all but a negligible fraction of $\mathcal{S}_2$'s randomness T_2, in the residual two-party protocol Π_{res},*

- *Either one of $\{\mathcal{A}_i^{<}(T_2)\}_{i\in[R]}$, $\{\mathcal{B}_i^{<}(T_2)\}_{i\in[R]}$ or $\mathcal{A}_0(T_2)$ biases the output to be smaller than b by a non-negligible amount,*

- *Or one of $\{\mathcal{A}_i^{\geq}(T_2)\}_{i\in[R]}$, $\{\mathcal{B}_i^{\geq}(T_2)\}_{i\in[R]}$, or $\mathcal{A}_0(T_2)$ biases the output to b or larger by a non-negligible amount.*

Proof. By T_2-equity property, for all but a negligible fraction of T_2, when everyone behaves honestly, the output of the residual protocol Π_{res} is indistinguishable from the uniform distribution. The bias towards output that is smaller than b that $\mathcal{A}_i^{<}(T_2)$ can cause is

$$\Pr[\alpha_i < b \wedge \beta_i < b] + \Pr[\alpha_i \geq b \wedge \beta_{i-1} < b] - \frac{b}{m}.$$

The bias towards output that is greater than or equal b that $\mathcal{A}_i^{\geq}(T_2)$ can cause is

$$\Pr[\alpha_i \geq b \wedge \beta_i \geq b] + \Pr[\alpha_i < b \wedge \beta_{i-1} \geq b] - \frac{m-b}{m}.$$

The bias towards output that is smaller than b that $\mathcal{B}_i^{<}(T_2)$ can cause is

$$\Pr[\beta_i < b \wedge \alpha_{i+1} < b] + \Pr[\beta_i \geq b \wedge \alpha_i < b] - \frac{b}{m}.$$

The bias towards output that is greater than or equal b that $\mathcal{B}_i^{\geq}(T_2)$ can cause is

$$\Pr[\beta_i \geq b \wedge \alpha_{i+1} \geq b] + \Pr[\beta_i < b \wedge \alpha_i \geq b] - \frac{m-b}{m}.$$

Let $\overline{\mathsf{bias}}$ denote the average of these $4R+1$ biases. Then

$$\begin{aligned}
\overline{\mathsf{bias}} = \frac{1}{4R+1} \Bigg[& \max\left\{\Pr[\beta_0 < b] - \frac{b}{m}, \Pr[\beta_0 \geq b] - \frac{m-b}{m}\right\} \\
& + \sum_{i=1}^{R} \Pr[\alpha_i < b \wedge \beta_i < b] + \Pr[\alpha_i \geq b \wedge \beta_{i-1} < b] - \frac{b}{m} \\
& \quad + \Pr[\alpha_i \geq b \wedge \beta_i \geq b] + \Pr[\alpha_i < b \wedge \beta_{i-1} \geq b] - \frac{m-b}{m} \\
& \quad + \Pr[\beta_i < b \wedge \alpha_{i+1} < b] + \Pr[\beta_i \geq b \wedge \alpha_i < b] - \frac{b}{m} \\
& \quad + \Pr[\beta_i \geq b \wedge \alpha_{i+1} \geq b] + \Pr[\beta_i < b \wedge \alpha_i \geq b] - \frac{m-b}{m} \Bigg]
\end{aligned}$$

Since $\Pr[\alpha_i < b \wedge \beta_i < b] + \Pr[\alpha_i \geq b \wedge \beta_i \geq b] + \Pr[\beta_i \geq b \wedge \alpha_i < b] + \Pr[\beta_i < b \wedge \alpha_i \geq b] = 1$ for any i, rearrange the terms, we get

$$\begin{aligned}
\overline{\mathsf{bias}} &= \frac{1}{4R+1}\left[1 - \Pr[\alpha_{R+1} \neq \beta_R] - \Pr[\alpha_1 = \beta_0] + \max\left\{\Pr[\beta_0 < b] - \frac{b}{m}, \Pr[\beta_0 \geq b] - \frac{m-b}{m}\right\}\right] \\
&= \frac{1}{4R+1}\left[1 - \max\{\Pr[\beta_0 < b], \Pr[\beta_0 \geq b]\} + \max\left\{\Pr[\beta_0 < b] - \frac{b}{m}, \frac{b}{m} - \Pr[\beta_0 < b]\right\}\right] \\
&\geq \frac{1}{4R+1} \min\left\{\frac{b}{m}, \frac{m-b}{m}\right\} \geq \frac{1}{m(4R+1)}.
\end{aligned}$$

Therefore, at least one of these adversaries can make a non-negligible bias. □

By the b-wolf-minion condition, none of these adversaries is able to bias the output towards any direction except for 0. Otherwise, we can construct an adversary that breaks the b-wolf-minion condition of Π.

Lemma 8.7. *For all but a negligible fraction of T_2, we have*

1. *At least one of $\{\mathcal{A}_i^{<}(T_2)\}_{i\in[R]}$ and $\{\mathcal{B}_i^{<}(T_2)\}_{i\in[R]}$, or $\mathcal{A}_0(T_2)$ can bias the output towards outcomes smaller then b.*

2. *None of these adversaries can bias the output towards b or larger by a non-negligible amount.*

Proof. Suppose that the second claim is not true. Then, there must exist an T_2^*, such that one of these adversaries $\mathcal{A}(T_2^*)$ can bias the output towards b or larger.

Now consider an adversary $\mathcal{A}^*$ that takes T_2^* as advice and corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$. It fixes $\mathcal{S}_2$'s randomness to T_2^* and follows the strategy of $\mathcal{A}$. By the assumption, $\mathcal{A}^*$ can bias the output towards b or larger by a non-negligible amount in Π, which contradicts the b-wolf-minion condition. Therefore, the second claim must be true. The first claim thus follows by combining Lemma 8.6 and the second claim. □

Now define adversaries $\bar{\mathcal{A}}_i^{<}$, $\bar{\mathcal{A}}_i^{\geq}$, $\bar{\mathcal{B}}_i^{<}$, $\bar{\mathcal{B}}_i^{\geq}$, and $\bar{\mathcal{A}}_0$ as follows.

- $\bar{\mathcal{A}}_i^{<}$ (resp. $\bar{\mathcal{A}}_i^{\geq}$) corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$, randomly picks T_2 for $\mathcal{S}_2$, and follows the strategy of $\mathcal{A}_i^{<}(T_2)$ (resp. $\mathcal{A}_i^{\geq}$).
- $\bar{\mathcal{B}}_i^{<}$ (resp. $\bar{\mathcal{B}}_i^{\geq}$) corrupts $\mathcal{S}_3$ and $\mathcal{S}_2$, randomly picks T_2 for $\mathcal{S}_2$, and follows the strategy of $\mathcal{B}_i^{<}(T_2)$ (resp. $\bar{\mathcal{B}}_i^{\geq}$).
- $\bar{\mathcal{A}}_0$ corrupts $\mathcal{S}_1$ and $\mathcal{S}_2$, randomly picks T_2 for $\mathcal{S}_2$, and follows the strategy of $\mathcal{A}_0(T_2)$.

By Lemma 8.7, at least one of $\bar{\mathcal{A}}_i^{<}$, $\bar{\mathcal{A}}_i^{\geq}$, $\bar{\mathcal{B}}_i^{<}$, $\bar{\mathcal{B}}_i^{\geq}$, and $\bar{\mathcal{A}}_0$ can bias the output towards outcomes smaller than b. However, an execution of Π interacting with $\bar{\mathcal{A}}_0$ is the same as an execution of Π interacting with an adversary corrupting only $\mathcal{S}_1$ and always aborts at the beginning of the protocol. According to the lone-wolf condition, $\bar{\mathcal{A}}_0$ should not be able to bias the output of Π towards any direction. Thus, it must be that

(Bias-C1) One of $\{\bar{\mathcal{A}}_i^{<}, \bar{\mathcal{B}}_i^{<}\}_{i\in[R]}$ can bias the output towards outcomes smaller than b;

(Bias-C2) None of $\{\bar{\mathcal{A}}_i^{\geq}, \bar{\mathcal{B}}_i^{\geq}\}_{i\in[R]}$ can bias the output towards b or larger.

Next, we show that conditions (Bias-C1) and (Bias-C2) cannot simultaneously hold due to the lone-wolf condition. This contradiction finishes the proof.

In the rest of this section, fix an arbitrary $i \in [R]$. We only give the proof for the sequence of $\bar{\mathcal{A}}_i$ adversaries since the same proof holds for $\bar{\mathcal{B}}_i$ adversaries. We say that $\bar{\mathcal{A}}_i^{<}$ causes μ-bias towards outcomes smaller than b if in an execution where $\mathcal{S}_3$ interacts with $\bar{\mathcal{A}}_i^{<}$, the probability $\Pr[\text{Output} < b] = \mu + \frac{b}{m}$.

Lemma 8.8. *For any $i \in [R]$, condition (Bias-C1) and (Bias-C2) cannot both be true. Specifically, if $\bar{\mathcal{A}}_i^{<}$ causes μ bias towards outcomes smaller than b, then $\bar{\mathcal{A}}_i^{\geq}$ causes at least $\mu - \mathsf{negl}(\lambda)$ bias towards outcomes greater than or equal to b.*

Proof. In an execution of Π, the randomness of three parties together defines a sample path. For $j = 0, \ldots, m-1$, let U denote the set of sample paths where $\bar{\mathcal{A}}_i^{\geq}$ aborts *before* sending the i-th round messages, and $\overline{U}$ denote the set of sample paths where $\bar{\mathcal{A}}_i^{\geq}$ aborts *after* sending the i-th round message. Then by definition, $\bar{\mathcal{A}}_i^{<}$ will abort *before* sending the i-th round messages on sample paths in U and *after* in $\overline{U}$.

Consider the following partition of the set U. Let $U_<^j$ denote the set of sample paths in U where $\mathcal{S}_3$'s outputs j when interacting with $\bar{\mathcal{A}}_i^{<}$, and $U_\geq^j$ denote the set of sample paths in U where

$\mathcal{S}_3$'s outputs j when interacting with $\bar{\mathcal{A}}_i^{\geq}$. Then $\{U_<^j\}_{j\in\{0,\ldots,m-1\}}$ (resp. $\{U_\geq^j\}_{j\in\{0,\ldots,m-1\}}$) forms a partition for U. We can define $\overline{U}_<^j$ and $\overline{U}_\geq^j$ analogously.

Now consider an adversary $\mathcal{A}^*_{\text{before}}$ that always aborts before sending the i-th round message. Then $\mathcal{A}^*_{\text{before}}$ can be viewed as an adversary that acts as $\bar{\mathcal{A}}_i^{\geq}$ on U and $\bar{\mathcal{A}}_i^{<}$ on $\overline{U}$. Note that $\mathcal{A}^*_{\text{before}}$ honestly chooses the randomness for $\mathcal{S}_2$ and always let $\mathcal{S}_1$ abort before sending the i-th round message. By the lone-wolf condition, it should not be able to bias the output towards any direction. Otherwise, an adversary controlling only $\mathcal{S}_1$ and always abort before sending the i-th round message can bias the output towards outcomes smaller than b, which contracts the lone-wolf condition.

Let N denote the total number of sample paths. Then, for any $j \in \{0,\ldots,m-1\}$

$$\frac{1}{N}(|U_\geq^j| + |\overline{U}_<^j|) = \frac{1}{m} \pm \mathsf{negl}(\lambda). \tag{9}$$

Consider an adversary $\mathcal{A}^*_{\text{after}}$ that always aborts after sending the i-th round message. By a similar argument as above, we get that

$$\frac{1}{N}(|U_<^j| + |\overline{U}_\geq^j|) = \frac{1}{m} \pm \mathsf{negl}(\lambda). \tag{10}$$

Observe that the bias that $\bar{\mathcal{A}}_i^{\geq}$ causes equals

$$\begin{aligned}
\frac{1}{N}\sum_{j\geq b}\left(U_\geq^j + \overline{U}_\geq^j\right) - \frac{m-b}{m} &\geq \sum_{j\geq b}\left(\frac{1}{m} - \frac{1}{N}U_<^j + \frac{1}{m} - \frac{1}{N}\overline{U}_<^j\right) - \frac{m-b}{m} - \mathsf{negl}(\lambda) \\
&= \frac{m-b}{m} - \Pr[\text{output} \geq b \text{ when interacting with } \bar{\mathcal{A}}_i^{<}] + \mathsf{negl}(\lambda) \\
&= \frac{m-b}{m} - \frac{m-b}{m} + \mu - \mathsf{negl}(\lambda) \\
&= \mu - \mathsf{negl}(\lambda).
\end{aligned}$$

□

9 Conclusion and Future Direction

In this work, we instantiate the comprehensive study of game-theoretic fairness for multi-party sampling from general distributions. We give an almost complete characterization for the case of uniform m-sided coin toss and protocols for any efficiently sampleable m-outcome distributions against majority-sized coalitions. While this work makes the initial effort to understand the theoretic landscape of game-theoretic fair distributed sampling, there are many interesting open questions left.

First, our protocol assumes a public preference profile, i.e., every player's preference is publicly known. In some real-world applications, players may want to hide their preferences in a distributed sampling task. Therefore, designing game-theoretic fair protocols for private preference profiles is an intriguing future work. In addition, the line of work exploring game-theoretic fairness [CGL$^+$18a, WAS22, CCWS21, KMSW22], including this work, has been focused mostly on the task of randomness generation. This leaves investigating game-theoretic fairness for more general functionalities (e.g., tasks where players want to compute some functions together based on everyone's input or tasks where players want to sample some correlated outputs together) as an important future direction.

References

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

[ADGH06a] 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.

[ADGH06b] Ittai Abraham, Danny Dolev, Rica Gonen, and Joseph 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.

[ADMM16] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and undefinedukasz Mazurek. Secure multiparty computations on bitcoin. *Commun. ACM*, 59(4):76–84, March 2016.

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

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

[ATM$^+$22] Lukas Aumayr, Sri AravindaKrishnan Thyagarajan, Giulio Malavolta, Pedro Moreno-Sanchez, and Matteo Maffei. Sleepy channels: Bi-directional payment channels without watchtowers. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 179–192, New York, NY, USA, 2022. Association for Computing Machinery.

[BHLT17] Niv Buchbinder, Iftach Haitner, Nissan Levi, and Eliad Tsfadia. Fair coin flipping: Tighter analysis and the many-party case. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2580–2600. SIAM, 2017.

[BHMO22] Amos Beimel, Iftach Haitner, Nikolaos Makriyannis, and Eran Omri. Tighter bounds on multiparty coin flipping via augmented weak martingales and differentially private sampling. *SIAM Journal on Computing*, 51(4):1126–1171, 2022.

[BHT18] Itay Berman, Iftach Haitner, and Aris Tentes. Coin flipping of any constant bias implies one-way functions. *Journal of the ACM (JACM)*, 65(3):1–95, 2018.

[BK14a] Iddo Bentov and Ranjit Kumaresan. How to use bitcoin to design fair protocols. In *CRYPTO*, 2014.

[BK14b] Iddo Bentov and Ranjit Kumaresan. How to use bitcoin to design fair protocols. In *CRYPTO*, pages 421–439, 2014.

[Blu83] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 15(1):23–27, jan 1983.

[BOO15] Amos Beimel, Eran Omri, and Ilan Orlov. Protocols for multiparty coin toss with a dishonest majority. *Journal of Cryptology*, 28(3):551–600, 2015.

[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 Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 3–32, Cham, 2021. Springer International Publishing.

[CGL$^+$18a] Kai-Min Chung, Yue Guo, Wei-Kai Lin, Rafael Pass, and Elaine Shi. Game theoretic notions of fairness in multi-party coin toss. In *Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part I*, page 563–596, Berlin, Heidelberg, 2018. Springer-Verlag.

[CGL$^+$18b] 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.

[Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 364–369, 1986.

[CMST22] Hao Chung, Elisaweta Masserova, Elaine Shi, and Sri AravindaKrishnan Thyagarajan. Rapidash: Foundations of side-contract-resilient fair exchange. Cryptology ePrint Archive, Paper 2022/1063, 2022. https://eprint.iacr.org/2022/1063.

[DEF18] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 967–984, New York, NY, USA, 2018. Association for Computing Machinery.

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

[DR07] Yevgeniy Dodis and Tal Rabin. Cryptography and game theory. In *AGT*, 2007.

[DSLMM11] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On the black-box complexity of optimally-fair coin tossing. In *Theory of Cryptography Conference*, pages 450–467. Springer, 2011.

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

[GKM$^+$13] Juan A. 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.

[Gol04] Oded Goldreich. *Foundations of Cryptography, Volume 2*. Cambridge university press Cambridge, 2004.

[GTZ15] Juan A. Garay, Björn Tackmann, and Vassilis Zikas. Fair distributed computation of reactive functions. In *DISC*, volume 9363, pages 497–512, 2015.

[HO14] Iftach Haitner and Eran Omri. Coin flipping with constant bias implies one-way functions. *SIAM Journal on Computing*, 43(2):389–409, 2014.

[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. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 408–416, 2014.

[IOZ14] Yuval Ishai, Rafail Ostrovsky, and Vassilis Zikas. Secure multi-party computation with identifiable abort. In *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II 34*, pages 369–386. Springer, 2014.

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

[KB14] Ranjit Kumaresan and Iddo Bentov. How to use bitcoin to incentivize correct computations. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 30–41. ACM, 2014.

[KMS$^+$16] Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 839–858. IEEE Computer Society, 2016.

[KMSW22] Ilan Komargodski, Shin'ichiro Matsuo, Elaine Shi, and Ke Wu. log*-round game-theoretically-fair leader election. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 409–438, Cham, 2022. Springer Nature Switzerland.

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

[KVV16] Ranjit Kumaresan, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Improvements to secure computation with penalties. In *ACM CCS*, 2016.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *Theory of Cryptography: Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. Proceedings 5*, pages 571–588. Springer, 2008.

[MGW87] Silvio Micali, Oded Goldreich, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*, pages 218–229. ACM New York, NY, USA, 1987.

[MNS16] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. *Journal of Cryptology*, 29(3):491–513, 2016.

[OPRV09] Shien Jin Ong, David C. Parkes, Alon Rosen, and Salil P. 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.

[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.

[WAS22] Ke Wu, Gilad Asharov, and Elaine Shi. A complete characterization of game-theoretically fair, multi-party coin toss. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 120–149, Cham, 2022. Springer International Publishing.

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