

Game-Theoretically Fair Coin Toss from Random Walk Against $n - 1$ Corruptions

Zirui Wang^{*1} and Ke Wu²

¹University of Illinois Urbana-Champaign, ziruiw8@illinois.edu

²University of Michigan, kewucse@umich.edu

Abstract

Coin-tossing protocols allow mutually distrustful parties to generate trusted randomness. While strong fairness is impossible against a corrupted majority, Chung et al. [CGL⁺18] introduced cooperative-strategy-proof (CSP) fairness for multi-party coin tossing, under the assumption that each party gets utility only when the outcome matches its public preference. CSP-fairness ensures that no PPT adversary can increase the expected joint utility of the corrupted parties through deviation. Since then, a line of work [WAS22, TSW24, ZW25] has explored the landscape of CSP-fair coin tossing and shown that CSP-fairness can be achieved even against a corrupted majority.

In this work, we study CSP-fair multi-sided coin tossing against up to $n - 1$ corruptions when parties may have arbitrary utilities over the possible outcomes. Perhaps surprisingly, we show that CSP-fairness against any semi-malicious adversary corrupting up to $n - 1$ parties is achievable for every full-row-rank utility matrix. Our construction can also be upgraded to fully malicious security. More generally, we introduce an abort-safe opening order condition on the utility matrix that suffices for resilience against $n - 1$ corruptions.

Our construction uses a public random walk whose state dynamically determines the distribution of the eventual outcome. The parties jointly generate randomness deciding each walk step using commit-and-reveal. When a party aborts, the protocol adjusts the transition distribution of the walk based on the parties' utilities. These adjustments, together with a carefully chosen opening order, prevent profitable deviations.

We also identify conditions on the utility matrix under which CSP-fair coin tossing is impossible. For three parties, our positive and negative results together give a complete characterization of the utility matrices for which CSP-fairness against two corruptions is achievable.

^{*}Much of this work was done while Zirui Wang was at the University of Michigan.

Contents

1	Introduction	3
1.1	Our Contribution	3
1.2	Related Work	5
2	Technical Overview	6
2.1	From Outcome Punishments to Distribution Punishments	7
2.2	The Random Walk Protocol	7
2.3	Formalizing Punishment Updates and the Opening Order	9
2.4	Why the Protocol Is CSP-Fair	12
2.5	Overview of the Impossibility Result	13
3	Model and Preliminaries	15
3.1	CSP-Fair Multi-Sided Coin Toss Protocol with Public Preference	15
3.2	Coalition-Utility Cones	16
4	The Random Walk Protocol	18
4.1	The Random Walk and Its Auxiliary Distribution	18
4.2	Protocol and Abort Handling	19
4.3	CSP-Fairness of Π_{RW}	23
5	Rank Condition for Maximal Resilience	28
6	Impossibility	30
A	Computing the Public Protocol Parameters	39
A.1	Opening Orders	39
A.2	Punishment Directions	41
A.3	Choosing the Walk Bias	44
A.4	Efficient Public Parameters	45
B	Rational Non-uniform Targets	45
C	More Examples and Complete Characterization for $n = 3$	46
C.1	Rank Condition is Not Necessary for Abort-Safe Opening-Order Condition	46
C.2	An Obstruction with $k = 3$	47
C.3	Complete Characterization for Three Parties	48

1 Introduction

In distributed randomness generation, or *coin toss* protocols, a group of mutually distrustful parties jointly generate a uniformly random coin value. Such primitives are fundamental building blocks used in Byzantine agreement and consensus [Rab83, CKS05], randomness beacons [SJKK⁺17, CD17], and secure computation [GMW87, Lin03]. Coin toss protocols have been studied in a long line of work in cryptography and distributed computing [Blu83, Cle86, BOO15, MNS16, BHLT17, BHMO22, Fei99, RZ01, RSZ02].

One important goal of coin toss protocols is *strong fairness*: the outcome should remain unbiased even if some parties deviate. Strong fairness can be achieved from standard secure multi-party computation [Yao82, GMW87, RBO89], assuming an honest majority. Unfortunately, Cleve’s celebrated impossibility [Cle86] showed that strong fairness is impossible against half corruption in the standard model (without timing assumptions). Yet the dishonest-majority regime is particularly relevant in decentralized settings, where many pseudonyms involved in the protocol may be controlled by a small number of strategic entities and an honest majority cannot always be assumed.

Game-Theoretic Fairness. In light of this, Chung, Guo, Lin, Pass, and Shi [CGL⁺18] model the adversary as a rational player who deviates to improve its utility. Under this view, they introduced *cooperative-strategy-proofness* (CSP-fairness). Imagine that each party has a publicly known preference for one of the two possible outcomes of a binary coin: heads or tails. Each party gets utility 1 if the final outcome matches their preference and 0 otherwise. A corrupted coalition’s utility is the sum of all colluding parties’ utilities. CSP-fairness guarantees that any non-uniform PPT adversary should not be able to improve its expected utility compared to an honest execution, except by a negligible amount. In other words, CSP-fairness ensures that the honest protocol forms a coalition-resistant Nash equilibrium. Interestingly, they gave a construction for binary coin toss that achieves CSP-fairness against a corrupted majority, and therefore circumvented Cleve’s impossibility result.

Since then, a line of work has developed CSP-fair coin toss protocols under different settings. In subsequent work, [WAS22] gave a complete characterization of the resilience threshold for binary coin toss as a function of parties’ public preferences, and [TSW24] then extended the construction to multi-sided coin toss, where each party gets utility 1 on one preferred outcome and 0 otherwise. More recently, [ZW25] study the setting where parties have a more general utility structure over different outcomes: instead of preferring a single outcome, a party may get different utilities from different outcomes. They analyze the resilience achievable under such richer utility structures.

Prior works reveal a fundamental dependence of the resilience threshold on parties’ utilities over different outcomes. With general utilities, this dependence becomes substantially more subtle: different coalitions may have very different, or even conflicting, preferences over the output, while a CSP-fair protocol must simultaneously ensure that none of the coalitions within the corruption threshold can profit from deviating. This motivates the question:

For general utilities, how far can we push the resilience of CSP-fair coin toss?

1.1 Our Contribution

Perhaps surprisingly, for a broad class of utility structures, we give a protocol that achieves CSP-fairness against $n - 1$ semi-malicious corruptions. This goes far beyond the best resilience previously known for multi-sided coin toss for a wide range of utility structures. Our construction takes a

different approach from prior work, using a random walk to dynamically evolve the distribution of the eventual outcome. We also give a complementary impossibility result that identifies utility structures for which CSP-fairness is impossible to achieve.

We first briefly introduce the model to describe our main results. Let λ denote the security parameter and $[n] = \{1, \dots, n\}$ be the n parties. The goal of the n parties is to toss a uniform random m -sided coin from $[m] = \{1, \dots, m\}$. Each party $i \in [n]$ has a rational utility vector $\mathbf{u}_i = (\text{util}_i(1), \dots, \text{util}_i(m))^\top \in \mathbb{Q}^m$, where $\text{util}_i(a)$ is party i 's utility when the outcome is a for $a \in [m]$. Let $U := (\mathbf{u}_1, \dots, \mathbf{u}_n)^\top \in \mathbb{Q}^{n \times m}$ be the publicly known utility matrix.

We consider a rushing, semi-malicious non-uniform PPT adversary $\mathcal{A}$ who corrupts a coalition $\mathcal{C}$ of size no more than t for some $t < n$. The coalition's utility is the sum of all coalition members' utilities. A semi-malicious adversary may program its randomness or abort early, but otherwise follows the protocol honestly. The protocol can be upgraded to withstand fully malicious adversaries using the GMW compilation paradigm [GMW87]. Since our construction uses only authenticated broadcast and its walk state is public, the compiler need only certify the commitments with concurrent zero-knowledge proofs of knowledge of the committed strings and opening randomness before each opening phase. Failed certifications and invalid openings are treated as aborts.

Random Walk Protocol. At a high level, our protocol maintains a random walk whose terminal state determines the final outcome. Each step of the walk is jointly generated through a commit-and-reveal procedure: All parties first broadcast commitments to their chosen randomness and then open these commitments. If all commitments are opened as prescribed, the walk takes the intended transition determined by the XOR of the committed randomness. If instead some party aborts and fails to open its commitment, the walk remains at its current state, while the protocol updates the transition distribution for the remainder of the walk based on the public utility matrix U and the cumulative set X of parties that have aborted. The update is chosen so that the abort does not increase the expected utility of *any* admissible coalition containing X , and hence of any admissible coalition that could make the observed aborts.

A key challenge is to ensure that the update is sufficient even when a rushing adversary has already learned the honest party's randomness in the opening phase before deciding whether to abort. For this reason, instead of having all parties open their commitments simultaneously in the same round, the protocol determines an *opening order* π for the remaining parties and asks them to open their commitments sequentially according to this order. The choice of π and the update rule ensure that aborting is never profitable.

Theorem 1.1 (Informal). *Fix n , m , and the utility matrix U . Assume that one-way functions exist. If the utility matrix U has full row rank, then the random walk protocol is a CSP-fair m -sided coin toss protocol against up to $n - 1$ corruptions.*

In fact, our random walk protocol achieves CSP-fairness for a wider family of utility matrices. The key requirement is that the parties need to be ordered so that every potential selective abort is handled in one of three ways: either (i) the coalition is indifferent among all outcomes; or (ii) the resulting update of the distribution strictly decreases any responsible coalition's expected utility; or (iii) the coalition must decide whether to abort before seeing the honest party's randomness that determines the current walk transition. We capture this property through the following *abort-safe opening-order condition*.

Definition 1.2 (Abort-safe opening order, informal). A utility matrix U satisfies the *abort-safe opening-order condition* at threshold t if, after every possible set X of parties that have aborted, the remaining parties can be ordered so that the following holds:

For every admissible coalition $\mathcal{C} \supseteq X$ of size no more than t , and every remaining member $i \in \mathcal{C} \setminus X$, either an abort by i allows the protocol to update the random walk distribution so as to strictly decrease $\mathcal{C}$'s expected utility, or some party outside $\mathcal{C}$ is scheduled to open after i .

Theorem 1.3 (Informal). *Fix n , m , and the utility matrix U . Assume that one-way functions exist. If the utility matrix U satisfies the abort-safe opening-order condition at threshold t , then the random walk protocol is a CSP-fair m -sided coin toss protocol against any t corruptions.*

We note that a utility matrix U of full row rank satisfies the abort-safe opening-order condition at threshold $n - 1$. But the opposite is generally not true. We give an example in Appendix C.1 of a utility matrix U with $\text{rank}(U) < n$ that still satisfies the abort-safe opening-order condition.

Impossibility. Our construction does not extend to every utility matrix. As a complementary result, we identify a class of utility matrices for which it is impossible to defend against t corruptions for $t \geq n/2$. Roughly speaking, for these utility matrices, CSP-fairness cannot hold simultaneously for all coalitions of size at most t : preventing profitable deviations by one collection of coalitions necessarily creates a profitable deviation for some other admissible coalitions.

To state the condition, for every coalition $\mathcal{C}$, let $\mathbf{u}_{\mathcal{C}} = \sum_{i \in \mathcal{C}} \mathbf{u}_i$ be the coalition's utility vector. Let $\mathbf{g}_{\mathcal{C}} := \sum_{i \in \mathcal{C}} (\mathbf{u}_i - \frac{1}{m} \langle \mathbf{u}_i, \mathbf{1} \rangle \mathbf{1})$ denote the coalition's centered aggregate utility vector so that its coordinates sum to zero. Here, $\mathbf{1}$ is the all-ones vector. Suppose the parties can be partitioned into

$$[n] = Y \cup P_1 \cup \dots \cup P_k,$$

for some $k \geq 2$ where every P_{ℓ} is nonempty. For each $\ell \in [k]$, let $A_{\ell} := \bigcup_{j \neq \ell} P_j$ and $C_{\ell} := [n] \setminus P_{\ell}$. Suppose there exists a vector $\mathbf{v}_{\ell} \in \mathbb{R}^m$ for each $\ell \in [k]$ such that: (i) $\mathbf{v}_{\ell}$ is a nonnegative linear combination of $\mathbf{g}_{\mathcal{C}}$ over admissible coalitions $\mathcal{C}$ containing C_{ℓ} ; and (ii) both $\mathbf{v}_{\ell}$ and $-\mathbf{v}_{\ell}$ are nonnegative linear combinations of $\mathbf{g}_{\mathcal{C}}$ over admissible coalitions $\mathcal{C}$ containing A_{ℓ} .

Theorem 1.4 (Impossibility, informal). *Fix n , m , $n/2 \leq t < n$, and the utility matrix U . If there exist a partition of $[n]$ and vectors $\mathbf{v}_{\ell}$ for $\ell \in [k]$ satisfying the above conditions with $|C_{\ell}| \leq t$ for every $\ell \in [k]$, and moreover,*

$$\frac{1}{m} \sum_{a \in [m]} \prod_{\ell=1}^k v_{\ell}(a) \neq 0,$$

then there is no CSP-fair coin toss protocol against t corruptions.

It is helpful to think of $\langle \mathbf{v}_{\ell}, \mathbf{q} \rangle$ as a score assigned to outcome distribution q . The conditions on $\mathbf{v}_{\ell}$ ensure that any noticeable bias in this score can be converted into a profitable deviation by some admissible coalition. Thus, under CSP-fairness, C_{ℓ} cannot noticeably increase the expected score, while A_{ℓ} cannot noticeably change it in either direction. Each score has mean zero under the uniform outcome, so a nonzero product average captures a dependence among the scores. We show that this dependence enables some admissible coalition to gain by choosing when to abort or by fixing its randomness. Together with Theorem 1.3, this gives a complete characterization for $n = 3$ and $t = 2$: CSP-fair three-party coin toss against 2 corruptions is possible if and only if the abort-safe opening-order condition holds (see Appendix C.3).

1.2 Related Work

Strongly Fair Coin Toss. The classical literature on fair coin tossing asks for strong fairness: the adversary should not be able to bias the outcome distribution in any direction by a non-negligible amount. Since Cleve's lower bound [Cle86], a long line of work studies the optimal

or near-optimal bias achievable in the dishonest-majority regime [BOO15, HT14, MNS16, AO16, BHLT17, BHMO22, DSLMM11]. CSP-fairness takes a different approach: the adversary is treated as rational and influences the output distribution to increase its expected utility. This relaxation is what makes fairness against a dishonest majority possible under utility structures.

Maximin Fairness. In addition to CSP-fairness, [CGL⁺18] also introduced *maximin fairness* as another game-theoretic relaxation of strong fairness. Maximin fairness protects the honest parties: no coalition should be able to decrease the expected utility of any honest party compared to an honest execution. In contrast, CSP-fairness cares about the incentive of the coalition itself. For binary coin toss, if the numbers of parties preferring the two outcomes are the same, the two notions coincide [CGL⁺18], whereas for unbalanced profiles they are distinct and lead to a different landscape [CGL⁺18]. Unfortunately, subsequent work of [WAS22] showed that maximin-fair binary coin toss is impossible against a corrupted majority.

Leader Election. Leader election asks n parties to elect one of themselves uniformly at random, or equivalently, to toss an n -sided coin. It has been studied extensively in the perfect-information model [Fei99, RZ01, RSZ02]. More recent work studies leader election under game-theoretic notions of fairness. [CCWS21] construct CSP-fair and maximin-fair leader election against an n -sized coalition under the preference assumption that each party only prefers itself being elected. They also gave sublogarithmic-round protocols achieving approximate game-theoretic fairness guarantees. Subsequent work [KMSW22] further obtained $O(\log^* n)$ -round protocols. [KKZ25] further study the possibility and round complexity of game-theoretically fair leader election, including lower bounds for achieving perfect fairness against an $(n - 1)$ -sized coalition.

Under the standard leader-election utilities considered in prior work, the corresponding utility matrix is the identity matrix $U = I_n$. CSP-fair leader election is therefore a special case of the model studied in our work, while leader election often additionally requires maximin fairness. In contrast to this line of work, which focuses on round complexity for the identity utility matrix, our work asks how the feasibility and impossibility of CSP-fairness change as a function of the utility matrix.

Organization. Section 2 gives a technical overview of the random walk protocol and the impossibility argument. Section 3 gives the formal model and preliminaries. Section 4 presents the random walk protocol and proves CSP-fairness under the abort-safe opening-order condition. Section 5 gives a sufficient rank condition for the abort-safe opening-order condition. Section 6 proves the impossibility result. Appendix A gives efficient algorithms for computing the public protocol parameters. Appendix B extends the construction to non-uniform coin toss. Appendix C provides more examples and the complete characterization for three parties.

2 Technical Overview

In this section, we explain the main idea behind our construction and impossibility. Throughout the overview, we will use a running example of a three-party three-sided coin toss with a utility matrix U

$$U = \begin{pmatrix} 3 & 1 & 2 \\ 0 & 2 & 4 \\ 2 & 4 & 0 \end{pmatrix} \tag{1}$$

and $t = 2$ to explain the intuition. Here, each row represents party i 's utility vector $\mathbf{u}_i$. For example, party 1 gets utility 3 on outcome 1, utility 1 on outcome 2 and utility 2 on outcome 3. A coalition $\mathcal{C}$'s joint utility vector $\mathbf{u}_{\mathcal{C}} = \sum_{i \in \mathcal{C}} \mathbf{u}_i$ is the sum of its coalition members' utility vectors. For example, if $\mathcal{C} = \{1, 2\}$, then $\mathbf{u}_{\mathcal{C}} = (3, 3, 6)$, which represents the coalition's joint utility on each outcome.

2.1 From Outcome Punishments to Distribution Punishments

The Approach in Prior Work. At a high level, prior CSP-fair coin toss constructions [CGL⁺18, WAS22, TSW24, ZW25] follow a common paradigm. The parties first commit to their local randomness and use secret sharing to share the committed randomness. If reconstruction succeeds, the jointly generated randomness determines the protocol outcome. The challenge arises when an adversary decides whether to abort to hamper the last reconstruction after learning enough information.

Prior constructions handle this selective abort by a default punishment outcome 0. If the adversary fails the final reconstruction, the protocol outputs 0 as a punishment. Therefore, prior work defends against coalitions whose joint utility on 0 is the worst among all possible outcomes.

This requirement is well suited to the structured preferences where each party only prefers one outcome among all m outcomes. However, it becomes restrictive for general utilities. Different potential coalitions may have completely different preferences and, in particular, different least-preferred outcomes. There may not exist a single outcome that can serve as a punishment for all admissible coalitions.

Our Approach: Distribution Punishment. We take a different approach. When an abort happens, we do not immediately collapse the protocol to a fixed punishment outcome. Instead, the protocol continues with a different target distribution for future rounds. The punishment is therefore a change in the continuation distribution over all outcomes rather than a predetermined final output. The challenge now becomes: how to pick a distribution punishment so that any admissible coalition is not incentivized to abort? This motivates the role of a random walk as defined below.

2.2 The Random Walk Protocol

Our protocol maintains a public random walk with a center 0 and m paths, one for each possible outcome $j \in [m]$. See Figure 1 for an example for $m = 3$. The random walk is parametrized with a transition probability $\alpha \in (1/2, 1)$. Its current state consists of a walk position w and a center distribution $\mathbf{p} = (p_1, \dots, p_m)$.

- The current walk position $w = (j, k)$ for $k \geq 1$ means the current position is at height k on the j -th path for $j \in [m]$. Height $k = 0$ means the center for all $j \in [m]$.
- A center distribution $\mathbf{p} = (p_1, \dots, p_m)$ over the m paths. From the center, the walk moves to position $(j, 1)$ with probability p_j .
- Once the walk is at position (j, k) on path j for some $k \geq 1$, the walk moves one step outward to $(j, k + 1)$ with probability α , and moves one step inward to $(j, k - 1)$ with probability $1 - \alpha$.

Figure 1: The random walk for three outcomes. At the center, $\mathbf{p}$ determines the path selected by the next transition. Once on a path, the walk moves outward with probability α and inward with probability $1 - \alpha$. Reaching height K on path j outputs j .

The walk starts at the center with $\mathbf{p} = \boldsymbol{\mu} = (1/m, \dots, 1/m)$ for the uniform distribution. Each step of the random walk, referred to as one invocation, is determined by randomness jointly generated by the parties. Every party maintains a set X of the parties that have aborted.

- **Commit phase:** Every party commits to a uniformly random string $x_i \leftarrow \{0, 1\}^\lambda$.
- **Open phase:** Parties open their commitments in a prescribed order π . If all openings succeed, the XOR of the committed strings samples the next prescribed transition.
- **Update $\mathbf{p}$:** Otherwise, if some party fails to open its commitment, add aborted parties to X . The walk position w remains unchanged, and the center distribution $\mathbf{p}$ is updated by $\mathbf{p} = \text{dist-update}(\mathbf{p}, X)$.

If the random walk reaches some height K on the j -th path, the protocol terminates and outputs j . Otherwise, the protocol continues for at most $T = \omega(\log \lambda)$ invocations and outputs a default outcome 1. In an honest execution, the distribution of the outcome is negligibly close to the uniform distribution.

Distribution Update dist-update and Opening Order π . We first explain the properties we want from the distribution update and the opening order, using the running example.

At the center, the role of $\mathbf{p}$ is especially simple: p_j is the probability of getting outcome j starting from the center. For intuition, we can think of updating $\mathbf{p}$ as directly updating the distribution of the final outcome. This is not literally true if the current walk position is not the center, but it captures the property we want from the punishment update. Later we formalize how changing $\mathbf{p}$ influences the eventual outcome distribution from an arbitrary walk position.

Suppose an opening failure occurs and let X be the cumulative set of parties that have aborted so far. The protocol does not know the actual corrupt coalition $\mathcal{C}$, but it knows that any coalition responsible for the observed aborts must contain X and have size at most t . Thus the update $\text{dist-update}(\mathbf{p}, X)$ must not increase the expected utility of *any* admissible coalition $\mathcal{C} \supseteq X$.

For example, suppose party 2 aborts, so $X = \{2\}$. The actual corrupt coalition could be $\{2\}$, $\{1, 2\}$, or $\{2, 3\}$, whose joint utility vectors are respectively

$$(0, 2, 4), \quad (3, 3, 6), \quad (2, 6, 4).$$

Consider an update $\mathbf{p}' = \mathbf{p} + \epsilon(1, 0, -1)$ for some small $\epsilon > 0$. The three possible coalitions' expected utilities change by -4ϵ , -3ϵ , and -2ϵ , respectively, under the new distribution $\mathbf{p}'$. Thus the same update strictly decreases the utility of every possible coalition that could have caused this abort. We call a distribution update with this property *safe* for X . The update itself should be small enough (the ϵ in the example) to ensure that the new $\mathbf{p}'$ remains a valid probability distribution.

If the adversary decides whether to abort before seeing all honest parties' randomness, the update above is already enough to ensure that all possible coalitions' expected utilities do not increase. The more difficult case is when the adversary waits till all honest parties open their randomness and then decides whether to abort. In this case, the adversary compares two expected utilities: finish this invocation and execute the committed transition, or abort and receive the

distribution punishment. In this case, simply decreasing the coalition's expected utility relative to the current state is not enough; the punishment must make aborting no better than executing the committed transition.

This is where the walk parameter α is useful. We want to make the utility loss of any single transition small enough for any possible coalition, so that avoiding the committed transition is not worth paying the punishment. For this purpose, we pick α sufficiently close to $1/2$; meanwhile, we need α to be strictly larger than $1/2$, so the walk eventually reaches a terminal outcome at height K with high probability.

However, a common strict punishment such as $\epsilon(1, 0, -1)$ does not always exist. Consider instead the case where party 1 aborts, so $X = \{1\}$. Consider the possible corrupt coalitions $\{1, 2\}$ and $\{1, 3\}$, whose utility vectors are

$$\mathbf{u}_{\{1,2\}} = (3, 3, 6), \quad \mathbf{u}_{\{1,3\}} = (5, 5, 2).$$

Their sum is a constant vector $(8, 8, 8)$. Therefore, for any distribution change, the utility changes of these two coalitions cancel out. Equivalently, any distribution change that strictly hurts one necessarily benefits the other. Consequently, any safe update must leave both coalitions indifferent. Such an indifferent punishment is not enough once the coalition has learned the committed transition: if the realized transition is unfavorable, the coalition would prefer to abort and switch to the indifferent punishment instead.

This is precisely why we need the opening order. Under $\pi = (1, 2, 3)$, party 1 opens first. Therefore, party 1 must decide whether to abort before seeing the committed transition. Before the transition is revealed, allowing the prescribed random transition to occur gives the coalition the same continuation utility in expectation as the current state. If the punishment update also leaves its continuation utility unchanged, aborting is therefore no better than continuing.

The construction is designed around this dichotomy. Whenever a coalition decides whether to abort after seeing all honest parties' randomness, the distribution update must make aborting no better than executing that committed transition. Whenever no such strict punishment is possible for all possible coalitions, the opening order must force that coalition to decide whether to abort before learning all honest parties' randomness.

2.3 Formalizing Punishment Updates and the Opening Order

We now formalize the two requirements from the previous subsection: 1.) a distribution update should simultaneously punish every coalition that could be responsible for the observed aborts; and 2.) whenever such a punishment may not be worse than the committed transition, the opening order should force the coalition to decide whether to abort before learning that transition.

Let $\mathbf{1} \in \mathbb{R}^m$ denote the all-ones vector. For a coalition $\mathcal{C}$, recall that $\mathbf{u}_{\mathcal{C}} = \sum_{i \in \mathcal{C}} \mathbf{u}_i$ is its joint utility vector. For a distribution $\mathbf{p}$ over the m outcomes, the coalition's expected utility when the outcome follows distribution $\mathbf{p}$ is

$$\overline{\text{util}}_{\mathcal{C}}(\mathbf{p}) := \langle \mathbf{p}, \mathbf{u}_{\mathcal{C}} \rangle.$$

For easier analysis, we center each party's utility vector by defining

$$\mathbf{g}_i := \mathbf{u}_i - \left(\frac{1}{m} \sum_{j=1}^m \text{util}_i(j) \right) \mathbf{1}, \quad \mathbf{g}_{\mathcal{C}} := \sum_{i \in \mathcal{C}} \mathbf{g}_i.$$

Then each $\mathbf{g}_{\mathcal{C}}$ has coordinates summing to zero. Moreover, for any two outcome distributions $\mathbf{p}$ and $\mathbf{p}'$, $\overline{\text{util}}_{\mathcal{C}}(\mathbf{p}') - \overline{\text{util}}_{\mathcal{C}}(\mathbf{p}) = \langle \mathbf{p}' - \mathbf{p}, \mathbf{g}_{\mathcal{C}} \rangle$. Thus $\mathbf{g}_{\mathcal{C}}$ captures exactly how coalition $\mathcal{C}$ evaluates a change in the outcome distribution.

In our running example, every row of U has average utility 2, so

$$\mathbf{g}_1 = (1, -1, 0), \quad \mathbf{g}_2 = (-2, 0, 2), \quad \mathbf{g}_3 = (0, 2, -2).$$

The two size-2 coalitions containing party 1 have centered utility vectors

$$\mathbf{g}_{\{1,2\}} = (-1, -1, 2), \quad \mathbf{g}_{\{1,3\}} = (1, 1, -2) = -\mathbf{g}_{\{1,2\}}.$$

Which Coalitions Are Punishable? Suppose the current cumulative abort set is X . The protocol collects the centered utility vectors of all coalitions that may be responsible for these aborts into

$$\mathcal{K}_t(X) := \left\{ \sum_{\mathcal{C} \supseteq X, 1 \leq |\mathcal{C}| \leq t} \beta_{\mathcal{C}} \mathbf{g}_{\mathcal{C}} : \beta_{\mathcal{C}} \geq 0 \right\}.$$

Thus $\mathcal{K}_t(X)$ consists of all nonnegative linear combinations of the centered utility vectors of coalitions that could be responsible for the observed aborts.

The distribution update rule updates $\mathbf{p}$ to some $\mathbf{p} + \mathbf{a}$. The distribution change $\mathbf{a}$ is safe for abort set X if it does not benefit any such coalition, namely, $\langle \mathbf{a}, \mathbf{g}_{\mathcal{C}} \rangle \leq 0$ for every $\mathcal{C} \supseteq X$ and $|\mathcal{C}| \leq t$. Equivalently, it suffices that

$$\langle \mathbf{a}, \mathbf{y} \rangle \leq 0 \quad \text{for every } \mathbf{y} \in \mathcal{K}_t(X).$$

For some coalitions $\mathcal{C}$, no distribution update can strictly decrease $\mathcal{C}$'s expected utility while remaining safe for every admissible coalition containing X . These are the coalitions $\mathcal{C}$ whose centered utility vector $\mathbf{g}_{\mathcal{C}}$ lies in the space

$$\mathcal{L}_t(X) := \mathcal{K}_t(X) \cap -\mathcal{K}_t(X),$$

where $-\mathcal{K}_t(X) = \{-\mathbf{y} : \mathbf{y} \in \mathcal{K}_t(X)\}$. We call $\mathcal{L}_t(X)$ the *unpunishable subspace*. Indeed, if $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(X)$, then both $\mathbf{g}_{\mathcal{C}}$ and $-\mathbf{g}_{\mathcal{C}}$ belong to $\mathcal{K}_t(X)$. Any safe distribution change $\mathbf{a}$ must therefore satisfy $\langle \mathbf{a}, \mathbf{g}_{\mathcal{C}} \rangle = 0$ for $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(X)$. Thus every safe punishment necessarily leaves coalition $\mathcal{C}$ indifferent.

Take the two cases above from our running example. If $X = \{2\}$, the possible coalitions have centered utility vectors

$$\mathbf{g}_{\{2\}} = (-2, 0, 2), \quad \mathbf{g}_{\{1,2\}} = (-1, -1, 2), \quad \mathbf{g}_{\{2,3\}} = (-2, 2, 0).$$

The direction $\mathbf{a} = \epsilon(1, 0, -1)$ has inner products $-4\epsilon, -3\epsilon, -2\epsilon$ with these three vectors, respectively. Hence all three possible coalitions can be strictly punished simultaneously by the distribution update $\mathbf{a}$.

In contrast, if $X = \{1\}$, both $\mathbf{g}_{\{1,2\}}$ and $\mathbf{g}_{\{1,3\}} = -\mathbf{g}_{\{1,2\}}$ belong to $\mathcal{K}_2(\{1\})$. Therefore,

$$\mathbf{g}_{\{1,2\}}, \mathbf{g}_{\{1,3\}} \in \mathcal{L}_2(\{1\}),$$

and every safe update must leave both coalitions indifferent. This is exactly the difficulty we observed in the previous subsection.

The Distribution Update Rule. A change to a probability distribution must preserve the sum of its coordinates. We therefore choose a vector $\mathbf{a}_X$ whose coordinates sum up to zero, such that

$$\begin{aligned} \langle \mathbf{a}_X, \mathbf{y} \rangle &= 0 \text{ for every } \mathbf{y} \in \mathcal{L}_t(X), \\ \text{and } \langle \mathbf{a}_X, \mathbf{y} \rangle &< 0 \text{ for every } \mathbf{y} \in \mathcal{K}_t(X) \setminus \mathcal{L}_t(X). \end{aligned}$$

Thus the same vector $\mathbf{a}_X$ weakly punishes every possible coalition containing X , and strictly punishes every such coalition whose utility direction is not inherently unpunishable.

We also choose $\mathbf{a}_X$ sufficiently small so that $\|\mathbf{a}_X\|_1 \leq 1/(2mt)$. This guarantees that repeated punishment updates always keep $\mathbf{p}$ as a valid distribution. The distribution update is then simply

$$\text{dist-update}(\mathbf{p}, X) := \mathbf{p} + \mathbf{a}_X.$$

The distribution update rule $\text{dist-update}(\cdot, \cdot)$ can be determined from U and t before the protocol begins. During the execution, parties simply feed the current center distribution $\mathbf{p}$ and abort set X to the rule to compute the update.

For the running example, set $\epsilon = 1/24$ and take

$$\begin{aligned} \mathbf{a}_{\{1\}} &= \epsilon(-1, 1, 0), \\ \mathbf{a}_{\{2\}} &= \mathbf{a}_{\{1,2\}} = \mathbf{a}_{\{2,3\}} = \epsilon(1, 0, -1), \\ \mathbf{a}_{\{3\}} &= \mathbf{a}_{\{1,3\}} = \epsilon(0, -1, 1). \end{aligned}$$

All six vectors have ℓ_1 norm $1/12 = 1/(2mt)$ and satisfy the required inequalities: only $\{1, 2\}$ and $\{1, 3\}$ are left indifferent when $X = \{1\}$.

Choosing the Walk Parameter α . As discussed before, strict punishment relative to the current distribution is not by itself sufficient once a rushing coalition has learned the committed transition.

The role of α is to make the utility effect of any single walk transition small enough to be dominated by the punishment. Before the protocol begins, we choose $\alpha > 1/2$ that works for every punishment vector that this rule may produce. Specifically, we require

$$\frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_C\|_1 \leq -\langle \mathbf{a}_X, \mathbf{g}_C \rangle \quad (2)$$

for every nonempty $X \subseteq \mathcal{C}$ with $|\mathcal{C}| \leq t$ and $\mathbf{g}_C \notin \mathcal{L}_t(X)$, where $\mathbf{a}_X$ denotes the punishment vector prescribed by the update rule for X . For every such pair, the right-hand side is strictly positive, while $(2\alpha - 1)/(1 - \alpha)$ approaches zero as α approaches $1/2$ from above. Thus one common α can be chosen to satisfy all these inequalities simultaneously.

For the running example, the minimum of $-\langle \mathbf{a}_X, \mathbf{g}_C \rangle / \|\mathbf{g}_C\|_1$ over these pairs is $1/48$. Taking $\alpha = 49/97$ gives $\frac{2\alpha - 1}{1 - \alpha} = 1/48$.

The analysis in the next subsection will show why (2) is needed. At a high level, it guarantees that, whenever $\mathcal{C}$ is strictly punishable, aborting after seeing the committed transition is no better than allowing that transition to occur.

The Abort-Safe Opening Order. It remains to pick an opening order to handle coalitions whose centered utility lies in the unpunishable subspace: such a coalition must make its last abort decision before learning the committed transition.

We enforce this through a global opening order π of the parties. Given a current abort set X , let $S := [n] \setminus X$ be the set of remaining active parties, and let π_S denote the order obtained by restricting π to the parties in S . For $i \in S$, let $\pi_S(i)$ denote party i 's position in this restricted order.

Consider an admissible coalition $\mathcal{C}$ with $X \subseteq \mathcal{C}$ and a remaining coalition member $i \in \mathcal{C} \setminus X$. If i aborts during the opening phase, the new cumulative abort set becomes $X \cup \{i\}$. We require

$$\mathbf{g}_C \in \mathcal{L}_t(X \cup \{i\}) \setminus \{\mathbf{0}\} \implies \exists h \in [n] \setminus \mathcal{C} \text{ such that } \pi_S(i) < \pi_S(h). \quad (3)$$

The premise says that, after i aborts, coalition $\mathcal{C}$ cannot be strictly punished by any safe distribution update because $\mathbf{g}_{\mathcal{C}}$ is in the unpunishable subspace given $X \cup \{i\}$. The conclusion guarantees that some party h outside the corrupt coalition is scheduled to open after i . Consequently, $\mathcal{C}$ cannot condition party i 's abort on the committed transition. We exclude $\mathbf{g}_{\mathcal{C}} = \mathbf{0}$ from the premise because such a coalition is indifferent among all outcome distributions and therefore has no incentive to selectively abort based on the outcome. We say that U satisfies the *abort-safe opening-order condition at threshold t* if there exists an opening order satisfying (3) for every possible abort set, admissible coalition, and remaining coalition member.

For the running example, take $\pi = (1, 2, 3)$. Consider first $X = \emptyset$ and coalition $\mathcal{C} = \{1, 2\}$. If party 1 aborts, then the new abort set is $\{1\}$, and $\mathbf{g}_{\{1,2\}} \in \mathcal{L}_2(\{1\})$ is unpunishable. But honest party 3 appears after 1, so party 1 must decide whether to abort before learning party 3's randomness. The same holds for $\mathcal{C} = \{1, 3\}$ since honest party 2 appears after party 1. These are exactly the two coalitions that could not be strictly punished after party 1 aborts.

By contrast, if party 1 reveals its secret and then party 2 aborts, all possible coalitions containing party 2 are strictly punishable by $\mathbf{a}_{\{2\}} = \epsilon(1, 0, -1)$. The opening-order condition therefore imposes no corresponding requirement that party 2 open before an honest party.

For rational U and fixed n, m , and t , Appendix A gives efficient algorithms for computing the update rule `dist-update`, a common rational α satisfying (2), and an abort-safe opening order π whenever one exists.

2.4 Why the Protocol Is CSP-Fair

We now explain why the punishment update and the abort-safe opening order imply CSP-fairness. In the previous subsections, we used $\mathbf{p}$ as an intuitive proxy for the distribution of the final outcome. The formal way to model this is an auxiliary distribution $\mathbf{q}$ associated with both the current center distribution $\mathbf{p}$ and the current walk position w .

Let $\rho := \frac{1-\alpha}{\alpha} \in (0, 1)$ and let $\mathbf{e}_j$ denote the distribution that outputs j with probability one. At the center, define $\mathbf{q}(\mathbf{p}, 0) := \mathbf{p}$, and at position (j, k) define

$$\mathbf{q}(\mathbf{p}, (j, k)) := (1 - \rho^k)\mathbf{e}_j + \rho^k\mathbf{p}.$$

For the corresponding infinite random walk with fixed $\mathbf{p}$, the auxiliary distribution $\mathbf{q}(\mathbf{p}, w)$ is exactly the distribution of the outcome on which the walk eventually escapes. Thus, $\mathbf{q}$ is the actual continuation-outcome distribution from an arbitrary walk position w . For our finite protocol, $\mathbf{q}$ will be negligibly close to the actual terminal outcome distribution once the walk reaches a sufficiently large height K .

The key property of $\mathbf{q}$ is that one prescribed random walk transition preserves it in expectation. Namely, if w' is sampled according to the prescribed transition from the current position w , then $\mathbb{E}[\mathbf{q}(\mathbf{p}, w')] = \mathbf{q}(\mathbf{p}, w)$. Consequently, before the committed transition is revealed, every coalition has the same expected utility from allowing this transition to occur as that at the current state.

The punishment update still works well with respect to $\mathbf{q}$. If the current position has height k and the protocol changes $\mathbf{p}$ to $\mathbf{p} + \mathbf{a}_X$, then

$$\mathbf{q}(\mathbf{p} + \mathbf{a}_X, w) - \mathbf{q}(\mathbf{p}, w) = \rho^k \mathbf{a}_X.$$

Therefore, all previous comparisons established for $\mathbf{a}_X$ continue to hold from any walk position, with both sides scaled by the same positive factor ρ^k . Consequently, from a position at height k , applying the punishment update changes coalition $\mathcal{C}$'s expected utility by $\rho^k \langle \mathbf{a}_X, \mathbf{g}_{\mathcal{C}} \rangle$, while the

absolute value of the utility change caused by any prescribed transition is at most

$$\rho^k \frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_C\|_1.$$

Thus multiplying (2) by ρ^k gives

$$\rho^k \frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_C\|_1 \leq -\rho^k \langle \mathbf{a}_X, \mathbf{g}_C \rangle.$$

In other words, the utility loss from the punishment is at least as large as the worst possible utility loss from executing one committed transition. Therefore, whenever the coalition is strictly punishable, aborting after learning the committed transition is no better than allowing that transition to occur.

We can now see how the two cases from the previous subsection cover every semi-malicious strategy. If a coalition can learn the committed transition before making its abort decision, the abort-safe opening-order condition (3) guarantees that the coalition is either indifferent among all outcomes or strictly punishable after the abort. In the strictly punishable case, the choice of α makes the punishment at least as bad as executing the committed transition.

On the other hand, if the coalition cannot be strictly punished, the opening order forces it to decide whether to abort while some honest parties have not opened their commitments. By commitment hiding, it cannot condition this decision on the realized transition; since a prescribed transition preserves $\mathbf{q}$ in expectation and the safe punishment does not increase its utility, aborting again provides no expected gain.

Thus, the adversary cannot increase its expected utility evaluated at the continuation distribution $\mathbf{q}$ in each invocation, except by a negligible amount from commitment hiding. Since initially $\mathbf{q} = \mathbf{p} = \boldsymbol{\mu}$, the coalition's expected utility throughout the execution remains at most its utility under the uniform distribution, up to a negligible error.

Finally, when the walk reaches height K on path M , $\mathbf{q}$ is within ρ^K of the point mass on M , which is negligible for sufficiently large K . Moreover, at most t invocations can be interrupted by aborts, and $\alpha > 1/2$ makes the probability of not reaching height K before a sufficiently large deadline T negligible. For the running example, $\rho = 48/49$, and we can take $K = \lambda$ and $T = 194(\lambda + 2)$, satisfying $K + t = (2\alpha - 1)T/2$ as required in Theorem 4.7. Thus the utility bound for $\mathbf{q}$ carries over to the actual protocol output, which gives us CSP-fairness.

2.5 Overview of the Impossibility Result

Our impossibility proof follows the Cleve-style approach used in prior work [CGL⁺18, WAS22], but adapts it to multi-sided outcomes with general utilities.

The Prior Lower-Bound Framework. The lower bound in [CGL⁺18, WAS22] concerns binary coin toss. They partition the parties into three super-parties S_1, S_2, S_3 and show that CSP-fairness implies three properties. The *lone-wolf* condition says that S_1 or S_3 alone cannot noticeably bias the binary outcome in either direction. The *wolf-minion* condition says that $S_1 \cup S_2$ or $S_2 \cup S_3$ cannot noticeably bias the outcome toward 1, which is their preferred outcome. Finally, the *rt₂-equity* condition says, roughly, that after fixing the randomness $\mathbf{rt}_2$ of the middle super-party S_2 , the honest outcome remains close to the uniform distribution for almost every $\mathbf{rt}_2$. Prior work shows that no protocol can satisfy all three properties simultaneously: after fixing $\mathbf{rt}_2$, the protocol becomes a residual two-party protocol between S_1 and S_3 . They then apply a sequence of Cleve-style abort adversaries to the residual protocol to prove the impossibility. [TSW24] extends this structure to multi-sided coin toss, but only for binary preferences.

We use a similar partition structure, but for general utilities there is no longer a canonical “preferred outcome”. For the overview, consider a partition

$$[n] = P_1 \cup Y \cup P_2,$$

where P_1, P_2 play the role of the two lone wolves, and Y plays the role of the middle super-party. We require $P_1 \cup Y$ and $P_2 \cup Y$ to have size at most the corruption threshold t whenever they are used as corrupt coalitions.

For multi-sided coin toss with a general utility structure, we choose two nonzero rational *score vectors*

$$\mathbf{v}_1 \in \mathcal{K}_t(P_1 \cup Y) \cap \mathcal{L}_t(P_1), \quad \mathbf{v}_2 \in \mathcal{K}_t(P_2 \cup Y) \cap \mathcal{L}_t(P_2).$$

For an outcome $j \in [m]$, we call $v_\ell(j)$ its $\mathbf{v}_\ell$ -score. Recall that $\mathbf{v} \in \mathcal{K}_t(\mathcal{C})$ is a nonnegative linear combination of the centered utility vectors $\mathbf{g}_D$ of admissible coalitions $D \supseteq \mathcal{C}$. Therefore, if an adversary controlling $\mathcal{C}$ noticeably increases the expected $\mathbf{v}$ -score, then at least one such coalition D obtains a noticeable utility gain by running the same attack by $\mathcal{C}$ while the additional corrupted parties $D \setminus \mathcal{C}$ behave honestly. By CSP-fairness, we get the two conditions below, which are analogues of the lone-wolf and wolf-minion conditions:

- An adversary controlling P_ℓ cannot noticeably increase or decrease the expected $\mathbf{v}_\ell$ -score. Otherwise, since $\mathbf{v}_\ell \in \mathcal{L}_t(P_\ell)$, both $\mathbf{v}_\ell$ and $-\mathbf{v}_\ell$ belong to $\mathcal{K}_t(P_\ell)$, so some admissible coalition obtains a noticeable utility gain. This is the analogue of the lone-wolf condition.
- An adversary controlling $P_\ell \cup Y$ cannot increase the expected $\mathbf{v}_\ell$ -score. Otherwise, it yields a profitable deviation.

The impossibility arises when the two score vectors are correlated under the target uniform outcome:

$$\eta := \left| \frac{1}{m} \sum_{j \in [m]} v_1(j) v_2(j) \right| > 0. \quad (4)$$

We show that when $\eta \neq 0$, we can build the sequence of Cleve-style abort adversaries that contradicts one of the above two conditions. Consider an R -round protocol. For each $\ell \in \{1, 2\}$ and round r , let $Z_{\ell,r}$ be the $\mathbf{v}_\ell$ -score of the outcome in the execution where P_ℓ follows the protocol through round r , sends its round- r message and then aborts. Couple all these executions using the same honest randomness and track $\Phi_r := \mathbb{E}[Z_{1,r} \cdot Z_{2,r}]$.

At $r = R$, both scores are evaluated on the same honest outcome, so correctness implies that $|\Phi_R|$ is noticeable since η is a constant independent of λ . We next consider the product Φ_0 arising from the two immediate-abort executions.

If $|\Phi_0|$ is noticeable, then for some $\ell \in \{1, 2\}$, the immediate-abort execution defining $Z_{\ell,0}$ has a noticeable score bias for a noticeable set of choices of Y ’s randomness. There are two cases:

- If this bias remains noticeable after averaging over the honest randomness of Y , then P_ℓ alone can noticeably increase or decrease the expected $\mathbf{v}_\ell$ -score by aborting at the beginning of the protocol. Since the honest expected $\mathbf{v}_\ell$ -score is negligible, this contradicts the first condition.
- Otherwise, the bias cancels when averaging over Y ’s randomness. There must exist some randomness $\mathbf{rt}_Y^*$ of Y such that $\mathbb{E}[Z_{\ell,0} \mid \mathbf{rt}_Y = \mathbf{rt}_Y^*]$ is noticeably positive. A semi-malicious adversary can then corrupt $P_\ell \cup Y$, program Y ’s randomness to $\mathbf{rt}_Y^*$, and make P_ℓ abort at the beginning of the protocol. This noticeably increases the expected $\mathbf{v}_\ell$ -score, contradicting the second condition.

Otherwise, suppose $|\Phi_0|$ is negligible. Since $|\Phi_R|$ is noticeable and the protocol has only polynomially many rounds, there must be some round r for which $|\Phi_r - \Phi_{r-1}|$ is noticeable. Since

$$\Phi_r - \Phi_{r-1} = \mathbb{E}[Z_{2,r}(Z_{1,r} - Z_{1,r-1}) + Z_{1,r-1}(Z_{2,r} - Z_{2,r-1})],$$

at least one of the two terms inside the expectation has noticeable expectation in absolute value. Without loss of generality, suppose $\varphi := |\mathbb{E}[Z_{2,r}(Z_{1,r} - Z_{1,r-1})]|$ is noticeable. Here, $Z_{1,r} - Z_{1,r-1}$ is exactly the change in the $\mathbf{v}_1$ -score between having P_1 send its round- r messages before aborting and having it abort before sending them.

We construct two adversaries. One adversary $\mathcal{B}$ controls only P_1 and honestly follows the protocol until round r . Then with probability $1/2$, it aborts before sending its round- r messages and with probability $1/2$, it aborts after sending its round- r messages. If $\mathcal{B}$ already has a noticeable expected $\mathbf{v}_1$ -score, it contradicts the first condition.

Otherwise, consider an adversary $\mathcal{B}'$ controlling $P_1 \cup Y$. After seeing P_2 's round- r messages, the adversary can compute $Z_{2,r}$ (i.e., assuming P_2 aborts) and use its value to decide whether P_1 should send its own round- r messages before aborting. Since φ is noticeable, it must be that $\mathcal{B}'$ noticeably increases the expected $\mathbf{v}_1$ -score relative to the execution with $\mathcal{B}$. Since the score in the execution with $\mathcal{B}$ is negligible, $\mathcal{B}'$ obtains a noticeably positive expected $\mathbf{v}_1$ -score, contradicting the second condition.

Putting all cases together, if $\eta > 0$, then no protocol is CSP-fair for such utility matrices U . The formal impossibility theorem extends the above argument from two lone wolves P_ℓ to an arbitrary number of P_ℓ , but the case above with P_1 and P_2 captures the main idea.

3 Model and Preliminaries

Notation. Let λ be the security parameter. Each party and the adversary are treated as non-uniform PPT algorithms. We use $[n]$ to denote the set $\{1, \dots, n\}$. Throughout this paper, $\mathbb{N}$ means the set of natural numbers, $\mathbb{N}^+$ means positive natural numbers, and $\mathbb{Q}$ means rational numbers.

Given the set of all possible outcomes $\Omega = [m]$, we use $\Delta(\Omega)$ to denote the set of all distributions over Ω . A distribution $\mathbf{q} \in \Delta(\Omega)$ is represented by its probability mass $\mathbf{q} = (q_1, \dots, q_m)$, where $q_j = \Pr_{M \leftarrow \mathbf{q}}[M = j]$ and $\sum_{j \in \Omega} q_j = 1$. For $j \in \Omega$, we use $\mathbf{e}_j$ to denote the unit vector where the j -th entry is one and all other entries are zero. We use $\mathbf{1}$ to denote the all-ones vector of length m and $\boldsymbol{\mu} = 1/m \cdot \mathbf{1}$ to denote the uniform distribution over Ω .

For a length- d vector $\mathbf{x}$, define the 1-norm $\|\mathbf{x}\|_1 := \sum_{i=1}^d |x_i|$. Unless otherwise stated, all vectors are column vectors.

3.1 CSP-Fair Multi-Sided Coin Toss Protocol with Public Preference

In an n -party m -sided coin toss protocol, n parties get together to jointly determine a uniform m -sided coin. We assume that n parties communicate in synchronous rounds over authenticated broadcast and authenticated private point-to-point channels. At the end of the protocol, parties agree on a single outcome $i \in [m]$. We require *public verifiability*: the final outcome can be computed from a deterministic function based on the broadcast messages throughout the execution.

We assume that the protocol proceeds in rounds. For most of this paper, we focus on a *rushing semi-malicious* adversary who corrupts a subset $\mathcal{C} \subsetneq [n]$ of the parties. In each round r , the adversary $\mathcal{A}$ waits for honest parties' messages and then decides corrupted parties' actions in round r . A *semi-malicious* adversary can program its randomness used in the protocol or abort early in the protocol, but otherwise follows the protocol honestly.

Each party i holds a public preference vector $\mathbf{u}_i = (\text{util}_i(1), \dots, \text{util}_i(m))$, where the j -th entry $\text{util}_i(j)$ denotes party i 's utility when the outcome is j . For a distribution $\mathbf{q} \in \Delta(\Omega)$, the expected utility of party i under outcome distribution $\mathbf{q}$ is

$$\overline{\text{util}}_i(\mathbf{q}) := \mathbb{E}_{j \leftarrow \mathbf{q}}[\text{util}_i(j)] = \langle \mathbf{q}, \mathbf{u}_i \rangle.$$

A coalition's utility is the sum of all coalition members' utilities. For a coalition $\mathcal{C} \subseteq [n]$, we define

$$\mathbf{u}_{\mathcal{C}} := \sum_{i \in \mathcal{C}} \mathbf{u}_i, \quad \overline{\text{util}}_{\mathcal{C}}(\mathbf{q}) := \sum_{i \in \mathcal{C}} \overline{\text{util}}_i(\mathbf{q}) = \langle \mathbf{q}, \mathbf{u}_{\mathcal{C}} \rangle.$$

The public preference matrix is

$$U := \begin{pmatrix} \mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_n^\top \end{pmatrix}.$$

We do not require any additional normalization condition on the utility matrix U . Since the preference matrix is publicly known and can be used in the protocol description, we write an n -party, m -sided coin toss with public preference matrix U as $\Pi_\lambda[U]$. When the context is clear, we may omit the security parameter λ .

Correctness. In an honest execution, the output should be a uniform m -sided coin. Formally,

Definition 3.1 (Correctness). Fix a public preference matrix U . A protocol $\Pi[U]$ is a correct n -party m -sided coin toss protocol if there exists a negligible function negl such that, for every security parameter λ ,

$$\left\| \text{out}_\lambda^{\Pi[U], H} - \boldsymbol{\mu} \right\|_1 \leq \text{negl}(\lambda),$$

where $\text{out}_\lambda^{\Pi[U], H}$ denotes the distribution of the outcome in an honest execution.

CSP-Fairness. We write $\mathcal{A}(\sigma)$ for $\mathcal{A}$ with arbitrary polynomial-length auxiliary input σ , included in its non-uniform advice. For an adversary $\mathcal{A}$ corrupting coalition $\mathcal{C}$, let $\text{out}_\lambda^{\Pi[U], \mathcal{A}}$ denote the outcome distribution of an execution of $\Pi[U]$ against adversary $\mathcal{A}$ while other parties behave honestly.

Definition 3.2 (CSP-fairness). Fix a public preference matrix U . A protocol $\Pi[U]$ is *cooperative-strategy-proof fair* (CSP-fair) against coalitions of size at most t if, for every nonempty coalition $\mathcal{C} \subsetneq [n]$ with $|\mathcal{C}| \leq t$ and every PPT adversary $\mathcal{A}$ controlling $\mathcal{C}$, there exists a negligible function negl such that for every λ ,

$$\overline{\text{util}}_{\mathcal{C}}\left(\text{out}_\lambda^{\Pi[U], \mathcal{A}(\sigma)}\right) \leq \overline{\text{util}}_{\mathcal{C}}\left(\text{out}_\lambda^{\Pi[U], H}\right) + \text{negl}(\lambda).$$

3.2 Coalition-Utility Cones

We introduce the geometric notation that will be used to describe the punishment direction so that selective aborting does not benefit any coalition containing the aborting party.

Distribution Directions. A change to a probability distribution must preserve the sum of the entries. We therefore work in the following zero-sum subspace

$$\mathcal{V}_m := \{\mathbf{a} \in \mathbb{R}^m : \langle \mathbf{a}, \mathbf{1} \rangle = 0\}.$$

Then for any $\mathbf{q} \in \Delta(\Omega)$ and $\mathbf{a} \in \mathcal{V}_m$, the vector $\mathbf{q} + \mathbf{a}$ is still a probability distribution if all its coordinates are nonnegative.

Since adding the same constant to all entries of a preference vector does not affect utility differences, or preferences, between distributions, it is convenient to consider the following centered vectors to represent expected utilities:

$$\mathbf{g}_i := \mathbf{u}_i - \overline{\text{util}}_i(\boldsymbol{\mu}) \cdot \mathbf{1}, \quad \mathbf{g}_C := \sum_{i \in C} \mathbf{g}_i = \mathbf{u}_C - \overline{\text{util}}_C(\boldsymbol{\mu}) \cdot \mathbf{1}. \quad (5)$$

These centered vectors belong to $\mathcal{V}_m$.

Claim 3.3. *For every coalition $C \subseteq [n]$ and any distributions $\mathbf{p}, \mathbf{q} \in \Delta(\Omega)$,*

$$\overline{\text{util}}_C(\mathbf{p}) - \overline{\text{util}}_C(\mathbf{q}) = \langle \mathbf{p} - \mathbf{q}, \mathbf{g}_C \rangle.$$

Proof. By linearity, the left-hand side equals $\langle \mathbf{p} - \mathbf{q}, \mathbf{u}_C \rangle$. Since $\langle \mathbf{p} - \mathbf{q}, \mathbf{1} \rangle = 0$, subtracting $\overline{\text{util}}_C(\boldsymbol{\mu}) \cdot \mathbf{1}$ from $\mathbf{u}_C$ leaves this inner product unchanged. $\square$

Thus, the sign of $\langle \mathbf{p} - \mathbf{q}, \mathbf{g}_C \rangle$ determines whether changing distribution from $\mathbf{q}$ to $\mathbf{p}$ is preferred by the coalition C or not. A positive sign means the coalition prefers $\mathbf{p}$ to $\mathbf{q}$ and a negative sign means the other way around. Zero means that the coalition has no preference over these two distributions.

Cones and Lineality. A convex cone is a nonempty set closed under addition and multiplication by nonnegative scalars. Formally, for a finite set of vectors $V \subseteq \mathbb{R}^m$, its *conic hull* consists of all nonnegative linear combinations:

$$\text{cone}(V) := \left\{ \sum_{\mathbf{v} \in V} \beta_{\mathbf{v}} \mathbf{v} : \beta_{\mathbf{v}} \geq 0 \right\}.$$

We use the convention $\text{cone}(\emptyset) = \{\mathbf{0}\}$. A cone generated by finitely many rational vectors is called a *rational polyhedral cone*.

For a cone $\mathcal{K}$, write $-\mathcal{K} := \{-\mathbf{y} : \mathbf{y} \in \mathcal{K}\}$ for its reflection through the origin. The intersection $\mathcal{K} \cap (-\mathcal{K})$ contains the directions that occur in $\mathcal{K}$ with both signs. It is the largest linear subspace contained in $\mathcal{K}$ and is called its *lineality space*. A cone is *pointed* if its lineality space is $\{\mathbf{0}\}$.

Coalition-Utility Cones. At any point of the protocol, those parties who have already aborted form a set A . We consider every possible coalition $C \supseteq A$ with $|C| \leq t$ that contains the set A . We need to update the random walk distribution so that any such possible coalition C does not prefer the new distribution. This is captured by the cone generated by these centered utility vectors and its lineality space.

Definition 3.4 (Coalition-utility cone and punishment direction). Given a threshold t , for every $A \subseteq [n]$, define

$$\begin{aligned} \mathcal{K}_t(A) &:= \text{cone}(\{\mathbf{g}_C : A \subseteq C \subseteq [n], 1 \leq |C| \leq t\}), \\ \mathcal{L}_t(A) &:= \mathcal{K}_t(A) \cap (-\mathcal{K}_t(A)). \end{aligned} \quad (6)$$

We call $\mathcal{K}_t(A)$ the *coalition-utility cone at A* and $\mathcal{L}_t(A)$ its lineality space. Both sets lie in $\mathcal{V}_m$. We call $\mathbf{a} \in \mathcal{V}_m$ a *punishment direction for A* if

$$\langle \mathbf{a}, \mathbf{y} \rangle \leq 0 \quad \text{for every } \mathbf{y} \in \mathcal{K}_t(A).$$

Because the cone consists of nonnegative combinations of the possible coalitions' centered utility vectors, this condition is equivalent to requiring that $\langle \mathbf{a}, \mathbf{g}_C \rangle \leq 0$ for every possible coalition C that contains A . Therefore, whenever $\mathbf{p}, \mathbf{p} + \mathbf{a}$ are both distributions over Ω , any such coalition does not prefer the new distribution $\mathbf{p} + \mathbf{a}$ compared to $\mathbf{p}$.

If a coalition's centered utility vector lies in $\mathcal{L}_t(A)$, then every punishment direction for A leaves that coalition's expected utility unchanged. In fact, if $\mathbf{g}_C \in \mathcal{L}_t(A)$, we cannot strictly decrease coalition C 's utility without increasing the utility of some other coalition containing A of size at most t . We therefore also call $\mathcal{L}_t(A)$ the *unpunishable subspace*.

Lemma 3.5 (Farkas' lemma [Sch86]). *Let $M \in \mathbb{R}^{d \times s}$ and $\mathbf{b} \in \mathbb{R}^d$. Exactly one of the following holds:*

1. *there exists $\mathbf{x} \in \mathbb{R}_{\geq 0}^s$ such that $M\mathbf{x} = \mathbf{b}$;*
2. *there exists $\mathbf{y} \in \mathbb{R}^d$ such that $M^\top \mathbf{y} \geq \mathbf{0}$ and $\langle \mathbf{y}, \mathbf{b} \rangle < 0$.*

4 The Random Walk Protocol

4.1 The Random Walk and Its Auxiliary Distribution

We first describe the random walk used in our protocol. The walk starts at a center called 0 and has m paths, one for each outcome $j \in \Omega$. Whenever the walk leaves the center, it chooses one of these paths according to a probability distribution $\mathbf{p} \in \Delta(\Omega)$. Once on a path, the walk moves one step away from the center with probability $\alpha > 1/2$ and one step toward the center with probability $1 - \alpha$ for some parameter α . If an inward step reaches the center, the walk chooses a new path according to $\mathbf{p}$ again. Eventually, if the walk reaches a position far enough away from the center, it determines the final outcome represented by that path. Our protocol implements this random walk using randomness sampled jointly from all n parties for each step.

Formally, fix a rational $\alpha \in (1/2, 1)$ and $\mathbf{p} = (p_1, \dots, p_m) \in \Delta(\Omega)$. The set of walk positions is

$$\mathcal{W} := \{0\} \cup (\Omega \times \mathbb{N}^+),$$

where 0 denotes the center and (j, k) denotes height k on path j . For convenience, we use the convention $(j, 0) := 0$ for every $j \in \Omega$. Let $(W_\ell)_{\ell \geq 0}$ denote the random walk starting from $W_0 = 0$. Its transition probabilities are defined as follows. From the center,

$$\Pr[W_{\ell+1} = (j, 1) \mid W_\ell = 0] = p_j \quad \text{for every } j \in \Omega.$$

From any position (j, k) with $k \geq 1$,

$$W_{\ell+1} = \begin{cases} (j, k+1), & \text{with probability } \alpha, \\ (j, k-1), & \text{with probability } 1 - \alpha. \end{cases}$$

The protocol does not run this random walk indefinitely. The walk terminates when it first reaches height K on any path. More formally, define

$$\tau_K := \inf\{\ell \geq 0 : W_\ell = (j, K) \text{ for some } j \in \Omega\}.$$

If $W_{\tau_K} = (j, K)$, the protocol terminates with outcome j . We use $W^\mathbf{p}$ to denote the current random walk transition based on center distribution $\mathbf{p}$.

Auxiliary Distribution $\mathbf{q}$. The distribution $\mathbf{p}$ is a public variable maintained by the protocol. It determines the probabilities with which the walk chooses its next path whenever it leaves the center. Initially $\mathbf{p} = \boldsymbol{\mu}$, and $\mathbf{p}$ remains fixed in an honest execution. Whenever an abort happens, the protocol updates $\mathbf{p}$ to another distribution but keeps the random walk position unchanged.

For the analysis, we define another auxiliary distribution $\mathbf{q}$ as follows. Let $\rho := \frac{1-\alpha}{\alpha} \in (0, 1)$. The auxiliary distribution $\mathbf{q}$ is defined with respect to a pair $(\mathbf{p}, w)$ for a random walk position w :

$$\begin{aligned}\mathbf{q}(\mathbf{p}, 0) &:= \mathbf{p}, \\ \mathbf{q}(\mathbf{p}, (j, k)) &:= (1 - \rho^k) \cdot \mathbf{e}_j + \rho^k \cdot \mathbf{p}.\end{aligned}\tag{7}$$

For a fixed $\mathbf{p}$, the auxiliary distribution $\mathbf{q}(\mathbf{p}, w)$ has an exact interpretation for the infinite walk defined above. Since $\alpha > 1/2$, the walk returns to the center only finitely many times with probability one, and hence there is almost surely a last time at which it leaves the center. We call the path chosen on this last departure the *eventual escape path*. The distribution $\mathbf{q}(\mathbf{p}, w)$ is exactly the distribution of the eventual escape path of this infinite walk when it starts from position w : Starting from the center, the eventual escape path is distributed according to $\mathbf{p}$; starting from (j, k) , the probability that the walk ever returns to the center is ρ^k ; otherwise its eventual escape path is j . Conditioned on returning to the center, the eventual escape path is again distributed according to $\mathbf{p}$.

This interpretation gives the key property for our analysis: a prescribed walk transition preserves $\mathbf{q}$ in expectation. The following property of $\mathbf{q}$ will be used throughout the security analysis.

Lemma 4.1 (Martingale property of the auxiliary distribution). *Fix $\mathbf{p} \in \Delta(\Omega)$ and suppose that one prescribed walk transition is taken while $\mathbf{p}$ is unchanged. For every current position $w \in \mathcal{W}$,*

$$\mathbb{E}[\mathbf{q}(\mathbf{p}, W_{\ell+1}) \mid W_\ell = w] = \mathbf{q}(\mathbf{p}, w).$$

Consequently, for every coalition $\mathcal{C} \subseteq [n]$,

$$\mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}(\mathbf{p}, W_{\ell+1})) \mid W_\ell = w] = \overline{\text{util}}_{\mathcal{C}}(\mathbf{q}(\mathbf{p}, w)).$$

Proof. For $w = 0$, the next position is $(j, 1)$ with probability p_j , and hence

$$\begin{aligned}\mathbb{E}[\mathbf{q}(\mathbf{p}, W_{\ell+1}) \mid W_\ell = 0] &= \sum_{j \in \Omega} p_j \mathbf{q}(\mathbf{p}, (j, 1)) \\ &= \sum_{j \in \Omega} p_j ((1 - \rho) \mathbf{e}_j + \rho \mathbf{p}) = (1 - \rho) \mathbf{p} + \rho \mathbf{p} = \mathbf{q}(\mathbf{p}, 0).\end{aligned}$$

For $w = (j, k)$ for some $k \geq 1$, the next position is $(j, k + 1)$ with probability α and $(j, k - 1)$ with probability $1 - \alpha$. By definition of $\rho = (1 - \alpha)/\alpha$, we have $\alpha \rho^{k+1} + (1 - \alpha) \rho^{k-1} = \rho^k$. It follows that

$$\begin{aligned}\mathbb{E}[\mathbf{q}(\mathbf{p}, W_{\ell+1}) \mid W_\ell = (j, k)] &= \alpha \mathbf{q}(\mathbf{p}, (j, k + 1)) + (1 - \alpha) \mathbf{q}(\mathbf{p}, (j, k - 1)) \\ &= (1 - \rho^k) \mathbf{e}_j + \rho^k \mathbf{p} = \mathbf{q}(\mathbf{p}, (j, k)).\end{aligned}$$

This proves the first claim. The second follows from the linearity of $\overline{\text{util}}_{\mathcal{C}}$. $\square$

4.2 Protocol and Abort Handling

We now implement the random walk using randomness jointly sampled by the parties. Fix a public preference matrix U and a corruption threshold $1 \leq t < n$. Write $\Pi_{\text{RW}}[U, K, T]$ for the protocol

with terminal height K and a deadline of T invocations over preference matrix U . When the context is clear, we may omit U from the notation. Both $K = K_\lambda$ and $T = T_\lambda$ will be chosen later as functions of the security parameter. Each invocation consists of one complete commit-and-reveal phase intended to generate a single walk transition. The deadline T is an upper bound on the total number of invocations, including those interrupted by abort.

Random Walk Protocol. We first describe our protocol and how the random walk is performed and prove the correctness. Throughout the execution, parties maintain a set X that records all aborted parties. Let π_S denote the prescribed opening order of the remaining active parties $S = [n] \setminus X$. We will present the opening order and the distribution update rule **dist-update** for handling abort afterwards.

Random Walk Protocol $\Pi_{\text{RW}}[U, K, T]$

Parameters: Initialize $\mathbf{p} = \boldsymbol{\mu}$, and the walk starts at $w = 0$. Initialize X as an empty set to denote the set of parties who have aborted. Parties maintain $(X, \mathbf{p}, w)$ throughout the protocol. For each invocation $\tau = 1, \dots, T$, parties do the following:

(Commit) Each alive party i samples a uniform random string $x_{i,\tau} \leftarrow \{0, 1\}^\lambda$ and fresh randomness $r_{i,\tau} \leftarrow \{0, 1\}^\lambda$ and broadcasts a commitment $c_{i,\tau} = \text{Comm}(x_{i,\tau}; r_{i,\tau})$. If any party k fails to commit, add k to X .

(Open) Parties broadcast their opening $(x_{i,\tau}, r_{i,\tau})$ according to the prescribed opening order π_S . If any party $k \notin X$ fails to open its commitment, add k to X , update $\mathbf{p} = \text{dist-update}(\mathbf{p}, X)$, and skip to the next invocation.

(Walk) Otherwise, if all commitments are successfully opened, compute $x_\tau = \bigoplus_{i \notin X} x_{i,\tau}$ and use x_τ to sample the prescribed walk transition of $W^{\mathbf{p}}$. Update w based on the sampled transition. If $w = (j, K)$ for some $j \in \Omega$, all parties terminate and output j .

If the protocol does not terminate by the end of invocation T , output the default value $j = 1$.

When the prescribed transition probabilities are not multiples of $2^{-\lambda}$, we round them to multiples of $2^{-\lambda}$ when sampling from x_τ ; this changes the distribution of each prescribed transition by at most $O(m \cdot 2^{-\lambda})$ in ℓ_1 distance.

Theorem 4.2 (Correctness). *If $T = \omega(\log \lambda)$ and $K \leq \frac{2\alpha-1}{2}T$, then $\Pi_{\text{RW}}[U, K, T]$ is correct.*

Proof. In an honest execution, no party aborts, so $X = \emptyset$ and $\mathbf{p} = \boldsymbol{\mu}$ throughout the protocol. Every invocation completes and each $x_\tau = \bigoplus_{i \in [n]} x_{i,\tau}$ is an independent uniformly random string.

First consider the finite walk in which these strings sample the prescribed transition probabilities exactly, and the walk is stopped either when it reaches height K or after T invocations.

Let k_τ denote the height of the walk after the τ -th invocation, and let τ_K be the first invocation at which the walk reaches height K . If the walk does not reach height K by the T -th invocation, set $\tau_K = \infty$. Let $Y_1, \dots, Y_T \in \{-1, 1\}$ be independent Bernoulli random variables such that $\Pr[Y_\tau = 1] = \alpha$ and $\Pr[Y_\tau = -1] = 1 - \alpha$. Whenever the walk is on a path, i.e., $k_{\tau-1} \geq 1$, we can use Y_τ as its height increment. Whenever the walk is at the center, its next height is always 1, so its height increment is $1 \geq Y_\tau$. Therefore, conditioned on the event $\tau_K > T$,

$$\sum_{\tau=1}^T Y_\tau \leq k_T < K.$$

Since $\mathbb{E}[Y_\tau] = 2\alpha - 1$, by Hoeffding's inequality and $K \leq (2\alpha - 1)T/2$, we have

$$\delta := \Pr[\tau_K > T] \leq \Pr\left[\sum_{\tau=1}^T Y_\tau < K\right] \leq \exp\left(-\frac{(2\alpha - 1)^2 T}{8}\right) = \text{negl}(\lambda),$$

where the last equality follows from $T = \omega(\log \lambda)$.

It remains to consider the distribution conditioned on $\tau_K \leq T$. The exact finite walk chooses a uniform path whenever it leaves the center, and its subsequent transition probabilities are independent of the path label. Moreover, the event $\{\tau_K \leq T\}$ is invariant under every permutation of the path labels. Therefore, for any $j, j' \in \Omega$,

$$\begin{aligned} \Pr[\tau_K \leq T \text{ and the terminal path is } j] \\ &= \Pr[\tau_K \leq T \text{ and the terminal path is } j'] \\ &= \frac{1 - \delta}{m}. \end{aligned}$$

Therefore, in this exact finite walk, the output probability distribution is $\frac{1-\delta}{m}$ for every $j \neq 1$ and $(1 - \delta)/m + \delta$ for $j = 1$, which is negligibly close to μ since δ is negligible.

Finally, with the actual uniform λ -bit randomness, each transition differs from the exact prescribed transition by at most $O(m2^{-\lambda})$ in ℓ_1 distance. Across all T invocations, the additional statistical distance is at most $O(mT2^{-\lambda})$, which is negligible because T is polynomially bounded. The theorem thus follows. $\square$

Distribution Updates dist-update after an Abort. We now introduce **dist-update** and compare coalition utility in the abort event. After an opening fails, X denotes the cumulative abort set. Every possible corrupt coalition therefore must contain X . The update of the center distribution $\mathbf{p}$ must not increase the expected utility of any such coalition. Moreover, since we consider a rushing adversary, we need to ensure that even after the coalition already sees the current x_τ , aborting and updating the distribution is no better than honestly opening.

We use the coalition-utility cones and punishment directions defined in Section 3. The following lemma gives a single direction that weakly punishes every compatible coalition and strictly punishes every compatible coalition outside the unpunishable subspace.

Lemma 4.3 (Punishment directions). *For every nonempty $X \subseteq [n]$ with $|X| \leq t$, there is a vector $\mathbf{a}_X \in \mathcal{V}_m$ with $\|\mathbf{a}_X\|_1 \leq 1/(2mt)$ such that*

$$\begin{aligned} \langle \mathbf{a}_X, \mathbf{y} \rangle &= 0 \quad \text{for every } \mathbf{y} \in \mathcal{L}_t(X), \\ \langle \mathbf{a}_X, \mathbf{y} \rangle &< 0 \quad \text{for every } \mathbf{y} \in \mathcal{K}_t(X) \setminus \mathcal{L}_t(X). \end{aligned}$$

In particular, $\langle \mathbf{a}_X, \mathbf{g}_C \rangle \leq 0$ for every $C \supseteq X$ with $|C| \leq t$.

Proof. Write $\mathcal{K} = \mathcal{K}_t(X)$ and $\mathcal{L} = \mathcal{L}_t(X)$. If $\mathcal{K} = \mathcal{L}$, take $\mathbf{a}_X = \mathbf{0}$. In the rest of the proof we focus on the case where $\mathcal{K} \supsetneq \mathcal{L}$.

Let $\mathcal{C}_1, \dots, \mathcal{C}_s$ be all compatible coalitions satisfying $X \subseteq \mathcal{C}_i$ and $|\mathcal{C}_i| \leq t$, and define the matrix $M = (\mathbf{g}_{\mathcal{C}_1}, \dots, \mathbf{g}_{\mathcal{C}_s})$. Then

$$\mathcal{K} = \{M\beta : \beta \geq \mathbf{0}\}.$$

Fix a compatible coalition $\mathcal{C}$ such that $\mathbf{g}_\mathcal{C} \notin \mathcal{L}$. It must be that $-\mathbf{g}_\mathcal{C} \notin \mathcal{K}$. Equivalently, there is no $\beta \geq \mathbf{0}$ satisfying $M\beta = -\mathbf{g}_\mathcal{C}$. By Farkas' lemma (Lemma 3.5), there exists $\mathbf{y}_\mathcal{C} \in \mathbb{R}^m$ such that $M^\top \mathbf{y}_\mathcal{C} \geq \mathbf{0}$ and $\langle \mathbf{y}_\mathcal{C}, -\mathbf{g}_\mathcal{C} \rangle < 0$.

Take $\mathbf{a}_{\mathcal{C}}$ as

$$\mathbf{a}_{\mathcal{C}} = -\mathbf{y}_{\mathcal{C}} + \frac{\langle \mathbf{y}_{\mathcal{C}}, \mathbf{1} \rangle}{m} \mathbf{1}.$$

Then $\mathbf{a}_{\mathcal{C}} \in \mathcal{V}_m$. Moreover, for every compatible coalition $\mathcal{C}'$, we have $\langle \mathbf{a}_{\mathcal{C}}, \mathbf{g}_{\mathcal{C}'} \rangle \leq 0$ while $\langle \mathbf{a}_{\mathcal{C}}, \mathbf{g}_{\mathcal{C}} \rangle < 0$ by definition of $\mathbf{y}_{\mathcal{C}}$ and linearity of inner products. Now define

$$\mathbf{a} := \sum_{\substack{\mathcal{C} \supseteq X, |\mathcal{C}| \leq t \\ \mathbf{g}_{\mathcal{C}} \notin \mathcal{L}}} \mathbf{a}_{\mathcal{C}}.$$

For every $\mathbf{y} \in \mathcal{K}$, each $\mathbf{a}_{\mathcal{C}}$ has nonpositive inner product with every generator $\mathbf{g}_{\mathcal{C}}$ of $\mathcal{K}$, and hence $\langle \mathbf{a}_{\mathcal{C}}, \mathbf{y} \rangle \leq 0$. Therefore, for every $\mathbf{y} \in \mathcal{K}$, we have $\langle \mathbf{a}, \mathbf{y} \rangle \leq 0$. Moreover, for any $\mathbf{y} \in \mathcal{K} \setminus \mathcal{L}$, let $\mathbf{y} = \sum_{i=1}^s \beta_i \mathbf{g}_{\mathcal{C}_i}$, where $\beta_i \geq 0$ for $i \in [s]$. Then there must exist some i with $\beta_i > 0$ and $\mathbf{g}_{\mathcal{C}_i} \notin \mathcal{L}$; otherwise $\mathbf{y}$ would be a linear combination of vectors in the subspace $\mathcal{L}$ and hence would itself belong to $\mathcal{L}$. For this $\mathcal{C}_i$,

$$\langle \mathbf{a}_{\mathcal{C}_i}, \mathbf{y} \rangle = \sum_{j=1}^s \beta_j \langle \mathbf{a}_{\mathcal{C}_i}, \mathbf{g}_{\mathcal{C}_j} \rangle < 0,$$

because every term is nonpositive and the $j = i$ term is strictly negative. All other terms in the definition of $\mathbf{a}$ have nonpositive inner product with $\mathbf{y}$, so

$$\langle \mathbf{a}, \mathbf{y} \rangle < 0.$$

Finally, if $\mathbf{y} \in \mathcal{L}$, then both $\mathbf{y}$ and $-\mathbf{y}$ belong to $\mathcal{K}$. Since $\langle \mathbf{a}, \mathbf{y} \rangle \leq 0$ and $-\langle \mathbf{a}, \mathbf{y} \rangle = \langle \mathbf{a}, -\mathbf{y} \rangle \leq 0$, we have $\langle \mathbf{a}, \mathbf{y} \rangle = 0$. Scaling $\mathbf{a}$ by a sufficiently small positive constant preserves all these inequalities and gives the desired vector $\mathbf{a}_X \in \mathcal{V}_m$ with $\|\mathbf{a}_X\|_1 \leq 1/(2mt)$. $\square$

For every nonempty X with $|X| \leq t$, fix the particular vector $\mathbf{a}_X$ specified by the public-data construction in Appendix A; these are the same vectors used when choosing the common parameter α below.

Define

$$\text{dist-update}(\mathbf{p}, X) := \mathbf{p} + \mathbf{a}_X. \tag{8}$$

These vectors depend only on the public preference matrix U (and therefore $\mathbf{g}_{\mathcal{C}}$ for all possible coalitions $\mathcal{C}$), the corruption threshold t , and the cumulative abort set X . For completeness, set $\mathbf{a}_X = \mathbf{0}$ when $X = \emptyset$ or $|X| > t$.

Every invocation that applies a punishment update adds a new party to X , so an execution against at most t corrupt parties applies at most t such updates. Starting from $\mathbf{p} = \boldsymbol{\mu}$, every coordinate therefore remains at least $\frac{1}{m} - \frac{t}{2mt} = \frac{1}{2m}$. Since every $\mathbf{a}_X$ belongs to $\mathcal{V}_m$, the coordinates continue to sum to one. Thus every reachable $\mathbf{p}$ is a probability distribution with positive coordinates.

Choosing the Walk Parameter α . It remains to choose the size of a random walk step relative to the punishment updates. Suppose a rushing coalition has learned the current x_τ and decides to abort. We need to make sure that such selective abort is not favorable compared to finishing the τ -th invocation. For this purpose, we choose the walk parameter α so that the utility loss from the punishment update is at least as large as the utility loss from *any* single prescribed walk transition. Specifically, we choose α such that

$$\frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_{\mathcal{C}}\|_1 \leq -\langle \mathbf{a}_X, \mathbf{g}_{\mathcal{C}} \rangle \tag{9}$$

for every nonempty $X \subseteq \mathcal{C}$ with $|\mathcal{C}| \leq t$ and $\mathbf{g}_\mathcal{C} \notin \mathcal{L}_t(X)$. Since the right-hand side is strictly positive for every such pair and the left-hand coefficient tends to zero as α approaches $1/2$ from above, such a common α exists. Algorithm 4 in Appendix A gives a polynomial-time construction of such an α .

The following lemma records the two consequences of the punishment update that we use in the fairness proof. The first follows from the direction of the update **dist-update** alone, whereas the second uses our choice of α .

Lemma 4.4 (Utility comparisons for a single punishment update). *Suppose an invocation τ failed with cumulative abort set X , changing the center distribution from $\mathbf{p}$ to $\mathbf{p}' = \text{dist-update}(\mathbf{p}, X)$ and keeping the walk at position w . Then, for every coalition $\mathcal{C} \supseteq X$ with $|\mathcal{C}| \leq t$,*

$$\overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}', w)) \leq \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w)). \quad (10)$$

If additionally $\mathbf{g}_\mathcal{C} \notin \mathcal{L}_t(X)$ or $\mathbf{g}_\mathcal{C} = \mathbf{0}$, then every position w' reachable by one prescribed transition from w with distribution $\mathbf{p}$ satisfies

$$\overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}', w)) \leq \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w')). \quad (11)$$

Proof. Let k be the height of w , with $k = 0$ at the center. By the definition of $\mathbf{q}$ and the centered-utility identity from Section 3.2,

$$\overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}', w)) - \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w)) = \rho^k \langle \mathbf{a}_X, \mathbf{g}_\mathcal{C} \rangle \leq 0,$$

where the last inequality follows from Lemma 4.3. This proves (10).

Next we prove Equation (11). The case $\mathbf{g}_\mathcal{C} = \mathbf{0}$ is immediate because the coalition is indifferent among all outcome distributions. Suppose $\mathbf{g}_\mathcal{C} \notin \mathcal{L}_t(X)$. At $w = (j, k)$ with $k \geq 1$, substituting the definition of $\mathbf{q}$ gives the outward and inward displacements

$$\begin{aligned} \mathbf{q}(\mathbf{p}, (j, k+1)) - \mathbf{q}(\mathbf{p}, (j, k)) &= \rho^k \frac{2\alpha - 1}{\alpha} (\mathbf{e}_j - \mathbf{p}), \\ \mathbf{q}(\mathbf{p}, (j, k-1)) - \mathbf{q}(\mathbf{p}, (j, k)) &= \rho^k \frac{2\alpha - 1}{1 - \alpha} (\mathbf{p} - \mathbf{e}_j). \end{aligned}$$

At the center, a transition to $(j, 1)$ changes the auxiliary vector by $\frac{2\alpha-1}{\alpha}(\mathbf{e}_j - \mathbf{p})$. Since $\alpha > 1 - \alpha$ and $|\langle \mathbf{p} - \mathbf{e}_j, \mathbf{g}_\mathcal{C} \rangle| \leq \|\mathbf{g}_\mathcal{C}\|_1$, in every case

$$|\overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w')) - \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w))| \leq \rho^k \frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_\mathcal{C}\|_1.$$

Combining this bound with (9) yields

$$\begin{aligned} \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}', w)) - \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w)) &\leq -\rho^k \frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_\mathcal{C}\|_1 \\ &\leq \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w')) - \overline{\text{util}}_\mathcal{C}(\mathbf{q}(\mathbf{p}, w)), \end{aligned}$$

which proves (11). $\square$

4.3 CSP-Fairness of Π_{RW}

Equation (10) shows that an abort never improves a coalition's utility evaluated at the auxiliary distribution $\mathbf{q}$ compared to the current state. However, if the coalition has already learned the committed x_τ and chooses to abort selectively, we need the stronger comparison (11): aborting must be no better than executing that particular transition for any x_τ . This stronger comparison is guaranteed only for coalitions whose centered utility vector $\mathbf{g}_\mathcal{C}$ lies outside the unpunishable subspace $\mathcal{L}_t(X)$.

The Opening-Order Condition. To address the above problem, we require the parties to open their commitments based on a specific order π_S for $S = [n] \setminus X$. The opening order is chosen so that at the time when the coalition learns enough information to compute the committed x_τ , any coalition that can still fail this invocation is either indifferent among all outcomes or is strictly punishable.

Given an ordering π_S and $i \in S$, let $\pi_S(i)$ denote the position of party i in that order. We require that, whenever an abort by a coalition member cannot strictly punish the coalition, at least one honest party is scheduled to open after that member.

Definition 4.5 (Abort-safe opening order). A public preference matrix U satisfies the *abort-safe opening-order condition at threshold t* if, for every $X \subseteq [n]$ with $|X| \leq t$, there exists an ordering π_S of $S = [n] \setminus X$ such that the following holds: For every coalition $\mathcal{C}$ satisfying $X \subseteq \mathcal{C}$ and $1 \leq |\mathcal{C}| \leq t$, and every party $i \in \mathcal{C} \setminus X$,

$$\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(X \cup \{i\}) \setminus \{\mathbf{0}\} \implies \exists h \in [n] \setminus \mathcal{C} : \pi_S(i) < \pi_S(h). \quad (12)$$

By Lemma A.1, whenever U satisfies this condition, there exists a single global order π of $[n]$ whose restriction $\pi_S := \pi|_S$ satisfies (12) for every abort set X ; we use these restricted orders throughout the protocol. We defer the efficient computation of the ordering π_S to the appendix. See Algorithm 2 in Appendix A.

Let $(\mathbf{p}_\tau, w_\tau)$ denote the public center distribution and walk position after invocation τ , and define $\mathbf{q}_\tau := \mathbf{q}(\mathbf{p}_\tau, w_\tau)$. Initially, $\mathbf{q}_0 = \boldsymbol{\mu}$. If the protocol terminates before invocation T , extend these variables through index T by keeping their terminal values unchanged.

Lemma 4.6. Fix n and m . Fix a public preference matrix U satisfying the opening-order condition at threshold t with opening orders π_S for any $X \subset [n]$, $|X| \leq t$. Consider $\Pi_{\text{RW}}[U, K, T]$ with *dist-update* defined in Equation (8) and α in Equation (9). Assume that the commitment scheme is computationally hiding and perfectly binding, and that K and T are polynomially bounded in λ . For every nonempty coalition $\mathcal{C} \subsetneq [n]$ with $|\mathcal{C}| \leq t$ and every PPT rushing semi-malicious adversary $\mathcal{A}$ controlling $\mathcal{C}$, there exists a negligible function negl such that

$$\mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_T)] \leq \overline{\text{util}}_{\mathcal{C}}(\boldsymbol{\mu}) + \text{negl}(\lambda). \quad (13)$$

Proof. Consider an execution of $\Pi_{\text{RW}}[U, K, T]$ in which an adversary $\mathcal{A}$ corrupts $\mathcal{C}$ and the remaining parties behave honestly.

One-Invocation Utility Comparison. Consider an invocation τ before termination. Let X be the abort set at the end of its commitment phase and $S = [n] \setminus X$. Since removing parties that failed to commit does not change $\mathbf{p}$ or w , at the end of the commitment phase, we have center distribution $\mathbf{p}_{\tau-1}$, walk position $w_{\tau-1}$, and auxiliary distribution $\mathbf{q}_{\tau-1}$.

Let x_τ^{com} be the XOR of the values uniquely bound at the end of the commitment phase, and let w_τ^{com} be the walk position from w selected by x_τ^{com} . Define $\mathbf{q}_\tau^{\text{com}} := \mathbf{q}(\mathbf{p}_{\tau-1}, w_\tau^{\text{com}})$. This is the auxiliary distribution that would result if the opening phase of invocation τ succeeded. By the binding property of commitments, this is well-defined.

Let h^* be the last honest party in the fixed global order π . Because only corrupted parties can be removed, h^* is also the last honest party in every restricted order π_S . Let E_τ denote the event that h^* successfully opens, i.e., any corrupted party before h^* in π_S does not abort.

If E_τ does not occur, the invocation fails. By Equation (10), $\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_\tau) \leq \overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_{\tau-1})$. If E_τ occurs, the last honest party h^* opens successfully and every party after h^* is corrupted. At this

time, the coalition learns x_τ^{com} and w_τ^{com} . If all successfully open their commitments, the invocation finishes honestly and $\mathbf{q}_\tau = \mathbf{q}_\tau^{\text{com}}$.

Otherwise, let i be the first party that fails to open. The opening-order condition gives $\mathbf{g}_C = \mathbf{0}$ or $\mathbf{g}_C \notin \mathcal{L}_t(X \cup \{i\})$. Equation (11) therefore gives $\overline{\text{util}}_C(\mathbf{q}_\tau) \leq \overline{\text{util}}_C(\mathbf{q}_\tau^{\text{com}})$. Combining the two cases,

$$\overline{\text{util}}_C(\mathbf{q}_\tau) - \overline{\text{util}}_C(\mathbf{q}_{\tau-1}) \leq \mathbf{1}_{E_\tau} (\overline{\text{util}}_C(\mathbf{q}_\tau^{\text{com}}) - \overline{\text{util}}_C(\mathbf{q}_{\tau-1})). \quad (14)$$

Hiding Hybrid. It remains to bound the expected right-hand side of (14). Define

$$Z_\tau^{\text{real}} := \mathbf{1}_{E_\tau} (\overline{\text{util}}_C(\mathbf{q}_\tau^{\text{com}}) - \overline{\text{util}}_C(\mathbf{q}_{\tau-1})),$$

and set this variable to zero if the protocol has already terminated before invocation τ .

Consider a hybrid that simulates the real execution through the beginning of invocation τ , samples independent uniform strings $x_{h^*,\tau}$ and $x'_{h^*,\tau}$, and replaces h^* 's commitment to $x_{h^*,\tau}$ with a commitment to $x'_{h^*,\tau}$. The simulation stops immediately before h^* opens, or at the first earlier opening failure, so it never needs to open the replaced commitment. Let S^{hyb} be the active set after the hybrid commitment phase, and define

$$\tilde{x}_\tau := x_{h^*,\tau} \oplus \bigoplus_{i \in S^{\text{hyb}} \setminus \{h^*\}} x_{i,\tau}^{\text{hyb}}.$$

Here $x_{i,\tau}^{\text{hyb}}$ is the value chosen for party i 's commitment in the hybrid, while $x_{h^*,\tau}$ is retained only for evaluating the score. Let $\tilde{\mathbf{q}}_\tau$ be the auxiliary distribution after applying the transition selected by $\tilde{x}_\tau$ to the pre-invocation state $(\mathbf{p}_{\tau-1}, w_{\tau-1})$, and put

$$Z_\tau^{\text{hyb}} := \mathbf{1}_{E_\tau^{\text{hyb}}} (\overline{\text{util}}_C(\tilde{\mathbf{q}}_\tau) - \overline{\text{util}}_C(\mathbf{q}_{\tau-1})).$$

As above, set this variable to zero when invocation τ is not reached.

In the hybrid, the retained string $x_{h^*,\tau}$ is independent of the simulated prefix, including S^{hyb} , the other committed values, and the event E_τ^{hyb} . With these quantities fixed, $\tilde{x}_\tau$ is therefore uniform. By Lemma 4.1 and the transition-rounding bound,

$$\left| \mathbb{E}[Z_\tau^{\text{hyb}}] \right| \leq O(m2^{-\lambda} \|\mathbf{g}_C\|_1).$$

Since $\mathcal{A}$ is semi-malicious, the reduction can simulate the randomness chosen by each corrupted party and hence knows the values $x_{i,\tau}$ to which corrupted parties commit; no extraction from the commitments is required. Thus both scores above are efficiently computable, while the challenged party h^* is fixed before the commitment phase and the reduction stops before its opening. Moreover, each score has absolute value at most $\|\mathbf{g}_C\|_1$. If their expectations differed non-negligibly, the reduction could output one with probability

$$\frac{1}{2} + \frac{Z}{2(1 + \|\mathbf{g}_C\|_1)},$$

where Z is the corresponding score, obtaining a non-negligible distinguisher for commitment hiding. Consequently,

$$\left| \mathbb{E}[Z_\tau^{\text{real}}] - \mathbb{E}[Z_\tau^{\text{hyb}}] \right| \leq \text{negl}(\lambda).$$

Combining the last two bounds gives $|\mathbb{E}[Z_\tau^{\text{real}}]| \leq \text{negl}(\lambda)$. Because T is polynomially bounded, summing these bounds over all invocations contributes only a negligible total error.

Taking expectations in (14) and summing over all T indices therefore gives

$$\mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_T)] - \mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_0)] = \sum_{\tau=1}^T \mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_{\tau}) - \overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_{\tau-1})] \leq \text{negl}(\lambda).$$

The lemma follows by taking $\mathbf{q}_0 = \boldsymbol{\mu}$. $\square$

Theorem 4.7 (CSP-fairness). *Fix n and m . Fix a public preference matrix U satisfying the opening-order condition at threshold t with opening orders π_S for any $X \subset [n]$, $|X| \leq t$. Consider $\Pi_{\text{RW}}[U, K, T]$ with **dist-update** defined in Equation (8) and α in Equation (9). Assume that the commitment scheme is computationally hiding and perfectly binding, and that K and T are polynomially bounded in λ and*

$$K = \omega(\log \lambda), \quad K + t \leq \frac{2\alpha - 1}{2}T.$$

Then $\Pi_{\text{RW}}[U, K, T]$ is CSP-fair against PPT rushing semi-malicious adversaries corrupting at most t parties.

Proof. Fix a nonempty coalition $\mathcal{C}$ with $|\mathcal{C}| \leq t$ and a PPT rushing semi-malicious adversary $\mathcal{A}$ controlling $\mathcal{C}$. Write $d := 2\alpha - 1 > 0$. Let k_{τ} be the height after invocation τ , and let τ_K be the first invocation at which the walk reaches height K , with $\tau_K = \infty$ if no such invocation occurs before the deadline.

Claim 4.8. $\Pr[\tau_K > T] \leq \text{negl}(\lambda)$ for some negligible function negl .

Proof of the claim. We first bound the probability that $\tau_K > T$. Let b_{τ} be the number of opening failures through invocation τ , and put $H_{\tau} := k_{\tau} + b_{\tau}$. Keep k_{τ} and b_{τ} unchanged after termination. Since every opening failure removes a new corrupted party, $b_{\tau} \leq t$.

For each invocation reached before termination, define $Y_{\tau}^{\text{com}} \in \{-1, 1\}$ from the committed XOR: on a path, set $Y_{\tau}^{\text{com}} = 1$ for an outward transition and $Y_{\tau}^{\text{com}} = -1$ for an inward transition. At the center, define Y_{τ}^{com} from the same committed XOR using any fixed efficiently computable Bernoulli partition with probability α of value 1, up to the same $O(2^{-\lambda})$ rounding error; this variable is used only for the analysis. A completed invocation increases the height by Y_{τ}^{com} on a path and by $1 \geq Y_{\tau}^{\text{com}}$ at the center. If the opening phase fails, the height remains unchanged and b_{τ} increases by one. Hence, whenever invocation τ starts before termination,

$$H_{\tau} \geq H_{\tau-1} + Y_{\tau}^{\text{com}}.$$

Let $Y_1, \dots, Y_T$ be independent reference variables with $\Pr[Y_j = 1] = \alpha$ and $\Pr[Y_j = -1] = 1 - \alpha$. For $s \in \{0, \dots, T\}$ and $z \in \mathbb{Z}$, define

$$F_s(z) := \Pr \left[\sum_{j=1}^s Y_j < K + t - z \right],$$

where an empty sum is zero. The function F_s takes values in $[0, 1]$, is nonincreasing in z , and satisfies

$$F_s(z) = \alpha F_{s-1}(z+1) + (1-\alpha)F_{s-1}(z-1) \quad (s \geq 1).$$

These probabilities are efficiently computable from s, z , and the rational parameter α . Define the bounded potential

$$V_{\tau} := \mathbf{1}_{\{\tau_K > \tau\}} F_{T-\tau}(H_{\tau}).$$

Monotonicity and the preceding height comparison give

$$V_\tau \leq \mathbf{1}_{\{\tau_K > \tau-1\}} F_{T-\tau}(H_{\tau-1} + Y_\tau^{\text{com}}).$$

The right-hand side is a bounded, efficiently computable score determined at the end of the commitment phase. Fix any honest party h . Apply the same hiding reduction as above, replacing h 's commitment by a commitment to an independent uniform string while retaining its original uniform string only for evaluating the hypothetical committed XOR and the score. The reduction stops at the end of the commitment phase, so it never needs to open the challenged commitment. Since $\mathcal{A}$ is semi-malicious, the reduction knows the randomness and committed values chosen for corrupted parties and therefore does not need to extract them from their commitments. In the hybrid, the retained string is independent of the adversary's view and of which corrupted parties fail to commit, so the hypothetical direction equals 1 with probability α , up to $O(2^{-\lambda})$ rounding error. Using the recurrence for F_s and commitment hiding therefore gives

$$\mathbb{E}[V_\tau] \leq \mathbb{E}[V_{\tau-1}] + \text{negl}(\lambda).$$

Because T is polynomially bounded, iterating this inequality yields

$$\mathbb{E}[V_T] \leq V_0 + \text{negl}(\lambda) = F_T(0) + \text{negl}(\lambda).$$

On the event $\{\tau_K > T\}$, we have $k_T < K$ and $b_T \leq t$, so $H_T < K + t$ and hence $F_0(H_T) = 1$. If the walk reaches height K by time T , the indicator in V_T is zero. Therefore $V_T = \mathbf{1}_{\{\tau_K > T\}}$, and

$$\begin{aligned} \Pr[\tau_K > T] &\leq \Pr\left[\sum_{j=1}^T Y_j < K + t\right] + \text{negl}(\lambda) \\ &\leq \exp\left(-\frac{d^2 T}{8}\right) + \text{negl}(\lambda) = \text{negl}(\lambda), \end{aligned}$$

where the last line uses $K + t \leq dT/2$ and $T = \omega(\log \lambda)$. $\square$

Next, we bound the difference between the coalition's actual utility and the auxiliary utility. Let M be the random variable denoting the protocol's output. There are two cases.

Case 1: $\tau_K \leq T$. The walk terminates at position (M, K) . Since the state remains unchanged after termination,

$$\mathbf{q}_T = \mathbf{q}(\mathbf{p}_T, (M, K)) = (1 - \rho^K) \mathbf{e}_M + \rho^K \mathbf{p}_T.$$

Therefore, $\mathbf{e}_M - \mathbf{q}_T = \rho^K (\mathbf{e}_M - \mathbf{p}_T)$. By claim 3.3,

$$|\text{util}_C(M) - \overline{\text{util}}_C(\mathbf{q}_T)| = |\langle \mathbf{e}_M - \mathbf{q}_T, \mathbf{g}_C \rangle| = \rho^K |\langle \mathbf{e}_M - \mathbf{p}_T, \mathbf{g}_C \rangle| \leq \rho^K \|\mathbf{g}_C\|_1,$$

where the last inequality follows because both $\mathbf{e}_M$ and $\mathbf{p}_T$ are probability distributions, so every coordinate of $\mathbf{e}_M - \mathbf{p}_T$ has absolute value at most one.

Case 2: $\tau_K > T$. On the event $\{\tau_K > T\}$, we use the trivial bound

$$|\text{util}_C(M) - \overline{\text{util}}_C(\mathbf{q}_T)| = |\langle \mathbf{e}_M - \mathbf{q}_T, \mathbf{g}_C \rangle| \leq \|\mathbf{g}_C\|_1.$$

Putting these bounds together, we have

$$\begin{aligned} \overline{\text{util}}_C\left(\text{out}_\lambda^{\Pi_{\text{RW}}[U, K, T], \mathcal{A}(\sigma)}\right) &= \mathbb{E}[\text{util}_C(M)] \\ &\leq \mathbb{E}[\overline{\text{util}}_C(\mathbf{q}_T)] + \rho^K \|\mathbf{g}_C\|_1 \Pr[\tau_K \leq T] \\ &\quad + \|\mathbf{g}_C\|_1 \Pr[\tau_K > T] \\ &\leq \mathbb{E}[\overline{\text{util}}_C(\mathbf{q}_T)] + \|\mathbf{g}_C\|_1 (\rho^K + \Pr[\tau_K > T]). \end{aligned}$$

By Lemma 4.6, $\mathbb{E}[\overline{\text{util}}_{\mathcal{C}}(\mathbf{q}_T)] \leq \overline{\text{util}}_{\mathcal{C}}(\boldsymbol{\mu}) + \text{negl}(\lambda)$. Moreover, $\rho < 1$ is a fixed constant and $K = \omega(\log \lambda)$, so ρ^K is negligible. We have also shown above that $\Pr[\tau_K > T]$ is negligible. Since U is fixed, $\|\mathbf{g}_{\mathcal{C}}\|_1$ is independent of λ . Hence

$$\overline{\text{util}}_{\mathcal{C}}\left(\text{out}_{\lambda}^{\Pi_{\text{RW}}[U, K, T], \mathcal{A}(\sigma)}\right) \leq \overline{\text{util}}_{\mathcal{C}}(\boldsymbol{\mu}) + \text{negl}(\lambda). \quad (15)$$

Since the coalition's expected utility in an honest execution is negligibly close to $\overline{\text{util}}_{\mathcal{C}}(\boldsymbol{\mu})$ by Theorem 4.2, we have

$$\overline{\text{util}}_{\mathcal{C}}\left(\text{out}_{\lambda}^{\Pi_{\text{RW}}[U, K, T], \mathcal{A}(\sigma)}\right) \leq \overline{\text{util}}_{\mathcal{C}}\left(\text{out}_{\lambda}^{\Pi_{\text{RW}}[U, K, T], H}\right) + \text{negl}(\lambda),$$

i.e., $\Pi_{\text{RW}}[U, K, T]$ is CSP-fair against coalitions of size at most t . $\square$

Appendix 5 proves that full row rank of U implies the abort-safe opening-order condition at threshold $t = n - 1$. With the parameter choices given there, Theorems 4.2 and 4.7 therefore yield Theorem 1.1.

5 Rank Condition for Maximal Resilience

We now give a sufficient condition on the public preference matrix U for the abort-safe opening-order condition in Definition 4.5. Recall the centered utility vectors $\mathbf{g}_i$ from (5), and define

$$G := (\mathbf{g}_1 \ \cdots \ \mathbf{g}_n) \in (\mathcal{V}_m)^n.$$

In particular, for every coalition $\mathcal{C}$, we have $\mathbf{g}_{\mathcal{C}} = G\mathbf{1}_{\mathcal{C}}$, where $\mathbf{1}_{\mathcal{C}}$ is an indicator vector with i -th entry being one if $i \in \mathcal{C}$ and zero otherwise. Write $\text{rank}(G)$ for the rank of G . We show that if G satisfies $\text{rank}(G) \geq n - 1$, or equivalently, the centered utility vectors have at most one linearly independent relation, then the utility matrix U satisfies the abort-safe opening-order condition.

Theorem 5.1. *Suppose that $\text{rank}(G) \geq n - 1$. There exists a single ordering π of $[n]$ such that, for every threshold $1 \leq t < n$ and every abort set $A \subseteq [n]$ with $|A| \leq t$, the ordering π_S on $S = [n] \setminus A$ obtained by deleting the parties in A from π is an abort-safe opening order (Definition 4.5).*

Proof. Recall that the kernel of G is $\ker(G) := \{\mathbf{x} \in \mathbb{R}^n : G\mathbf{x} = \sum_{i=1}^n x_i \mathbf{g}_i = \mathbf{0}\}$. By the rank assumption, the dimension of the kernel is $\dim \ker(G) \leq 1$. Hence $\ker(G)$ is either trivial or one-dimensional.

We first define an order π using only G . If $\ker(G)$ contains a nonzero nonnegative vector, choose a nonzero nonnegative generator $\boldsymbol{\kappa}$ of $\ker(G)$ and order the parties by nonincreasing κ_i , breaking ties by their indices. Otherwise, use the index order.

Fix any threshold $1 \leq t < n$, any abort set $A \subseteq [n]$ with $|A| \leq t$, any coalition $\mathcal{C}$ with $A \subseteq \mathcal{C}$ and $1 \leq |\mathcal{C}| \leq t$, and any party $i \in \mathcal{C} \setminus A$. Let $S := [n] \setminus A$ and $D := A \cup \{i\}$. Suppose that $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(D) \setminus \{\mathbf{0}\}$. We first show that there must exist a nonzero nonnegative vector in $\ker(G)$.

Since $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(D) = \mathcal{K}_t(D) \cap (-\mathcal{K}_t(D))$, we have $-\mathbf{g}_{\mathcal{C}} \in \mathcal{K}_t(D)$. By the definition of $\mathcal{K}_t(D)$, there exist coefficients $\theta_{\mathcal{C}'} \geq 0$ such that

$$-\mathbf{g}_{\mathcal{C}} = \sum_{\substack{D \subseteq \mathcal{C}' \subseteq [n] \\ |\mathcal{C}'| \leq t}} \theta_{\mathcal{C}'} \mathbf{g}_{\mathcal{C}'}.$$

For every coalition $\mathcal{C}'$, recall that $\mathbf{g}_{\mathcal{C}'} = G\mathbf{1}_{\mathcal{C}'}$. Define

$$\mathbf{b} := \sum_{\substack{D \subseteq \mathcal{C}' \subseteq [n] \\ |\mathcal{C}'| \leq t}} \theta_{\mathcal{C}'} \mathbf{1}_{\mathcal{C}'}, \quad s := \sum_{\substack{D \subseteq \mathcal{C}' \subseteq [n] \\ |\mathcal{C}'| \leq t}} \theta_{\mathcal{C}'}.$$

Then

$$G\mathbf{b} = \sum_{\substack{D \subseteq \mathcal{C}' \subseteq [n] \\ |\mathcal{C}'| \leq t}} \theta_{\mathcal{C}'} G\mathbf{1}_{\mathcal{C}'} = -\mathbf{g}_{\mathcal{C}}.$$

The vector $\mathbf{b}$ has a nice structure: every coalition $\mathcal{C}'$ appearing in the sum contains D . Therefore, for every $j \in D$, the j -th entry of $\mathbf{b}$ satisfies

$$b_j = \sum_{\substack{D \subseteq \mathcal{C}' \subseteq [n] \\ |\mathcal{C}'| \leq t}} \theta_{\mathcal{C}'} = s.$$

For every $j \notin D$, only some of the coalitions $\mathcal{C}'$ may contain j , and therefore $0 \leq b_j \leq s$.

Now define

$$\mathbf{v} := \mathbf{1}_{\mathcal{C}} + \mathbf{b} \geq \mathbf{0}.$$

Note that $\mathbf{v}$ is nonzero because $i \in \mathcal{C}$ gives $v_i \geq 1$. Using $G\mathbf{1}_{\mathcal{C}} = \mathbf{g}_{\mathcal{C}}$ and $G\mathbf{b} = -\mathbf{g}_{\mathcal{C}}$, we obtain

$$G\mathbf{v} = G\mathbf{1}_{\mathcal{C}} + G\mathbf{b} = \mathbf{g}_{\mathcal{C}} - \mathbf{g}_{\mathcal{C}} = \mathbf{0}.$$

Thus $\mathbf{v} \in \ker(G) \setminus \{\mathbf{0}\}$ is nonnegative. Consequently, the construction of π above used a nonnegative generator $\boldsymbol{\kappa}$ of $\ker(G)$. Since $\dim \ker(G) \leq 1$, we have $\mathbf{v} = \beta \boldsymbol{\kappa}$ for some $\beta > 0$. Therefore the fixed order π also orders the coordinates of $\mathbf{v}$ nonincreasingly.

We now show that this order π satisfies the abort-safe opening-order condition, i.e., there must exist an honest party h that is after i in the order π . Since $i \in D \subseteq \mathcal{C}$, we have $b_i = s$ and $(\mathbf{1}_{\mathcal{C}})_i = 1$, so

$$v_i = 1 + s.$$

On the other hand, for every $h \in [n] \setminus \mathcal{C}$, we have $(\mathbf{1}_{\mathcal{C}})_h = 0$ and $b_h \leq s$, so $v_h = b_h \leq s$. Consequently,

$$v_i = 1 + s > s \geq v_h \quad \text{for every } h \in [n] \setminus \mathcal{C}.$$

By construction, π orders parties by nonincreasing coordinates in $\mathbf{v}$, and hence i appears before every party outside $\mathcal{C}$.

Finally, because $i \notin A$ and $A \subseteq \mathcal{C}$, both i and every party in $[n] \setminus \mathcal{C}$ belong to $S = [n] \setminus A$. Restricting π to S therefore preserves their relative order $\pi_S(i) < \pi_S(h)$ for every $h \in [n] \setminus \mathcal{C}$. Since $t < n$, at least one such party h exists. This proves (12). Since t , A , $\mathcal{C}$, and i were all arbitrarily chosen, the same global order π satisfies the opening-order condition for every threshold and every abort history. $\square$

For rational U , this order can be computed by Gaussian elimination and sorting in polynomial time. We do not need to enumerate all possible coalitions to compute such a π . We defer the efficient algorithm for computing π from the utility matrix U to Appendix A.

Theorem 5.2. *Assume that the commitment scheme is computationally hiding and perfectly binding. Fix n , m , and a rational public preference matrix U such that $\text{rank}(U) = n$. There exist efficiently computable parameters K_λ, T_λ such that the random walk protocol $\Pi_{\text{RW}}[U, K_\lambda, T_\lambda]$ is a correct m -sided coin toss protocol that is CSP-fair against every PPT rushing semi-malicious adversary corrupting at most $n - 1$ parties.*

Proof. Set $t = n - 1$. Suppose that U has full row rank, and recall that its rows are $\mathbf{u}_1^\top, \dots, \mathbf{u}_n^\top$. Let

$$P := I_m - \frac{1}{m} \mathbf{1} \cdot \mathbf{1}^\top.$$

Since $\mathbf{g}_i = P\mathbf{u}_i$ for every $i \in [n]$, the centered utility matrix satisfies $G = PU^\top$.

Claim 5.3. $\text{rank}(G) \geq n - 1$.

Proof. The map P is the projection onto $\mathcal{V}_m$, and $\ker(P) = \text{span}\{\mathbf{1}\}$, which is one-dimensional. Let

$$W := \text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_n\} \subseteq \mathbb{R}^m.$$

Then $\dim W = \text{rank}(U)$, while

$$\text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_n\} = P(W) := \{P\mathbf{w} : \mathbf{w} \in W\}.$$

Applying rank-nullity to the restriction of P to W gives

$$\text{rank}(G) = \dim P(W) = \dim W - \dim(W \cap \ker(P)).$$

Since $\ker(P)$ is one-dimensional, $\dim(W \cap \ker(P)) \leq 1$. Therefore, $\text{rank}(G) \geq \text{rank}(U) - 1$. $\square$

It remains to instantiate the other public parameters, including the update rule Equation (8) and α in Equation (9) required by Π_{RW} . By Proposition A.6, all of these parameters can be obtained efficiently from the rational instance (n, m, t, U) .

Choose

$$K_\lambda := \left\lceil (\log_2(\lambda + 2))^2 \right\rceil \quad \text{and} \quad T_\lambda := \left\lceil \frac{2(K_\lambda + n - 1)}{2\alpha - 1} \right\rceil.$$

Because $2\alpha - 1 > 0$ is a fixed constant, both K_λ and T_λ are efficiently computable and polynomially bounded in λ . Moreover, both are $\omega(\log \lambda)$, and

$$\begin{aligned} K_\lambda + t &= K_\lambda + n - 1 \leq \frac{2\alpha - 1}{2} T_\lambda. \\ K_\lambda &\leq \frac{2\alpha - 1}{2} T_\lambda. \end{aligned}$$

Therefore the conditions of Theorem 4.2 and Theorem 4.7 are satisfied at threshold $t = n - 1$, so the protocol is correct and CSP-fair against every PPT rushing semi-malicious adversary corrupting at most $n - 1$ parties.

Finally, each invocation and each computation of the public update data takes polynomial time, and the protocol performs at most the polynomially bounded number T_λ of invocations. Hence $\Pi_{\text{RW}}[U, K_\lambda, T_\lambda]$ is polynomial time. $\square$

6 Impossibility

We now give a condition on the preference matrix under which no polynomial-round protocol can be CSP-fair against semi-malicious adversaries. Our argument extends the approach of [WAS22]: we first identify the lone-wolf and wolf-minion conditions implied by CSP-fairness, and then show that some semi-malicious strategy must violate one of these two conditions. The main difference is that [WAS22] track the condition based on a single binary outcome $b \in \{0, 1\}$, whereas we track utility scores as explained in Section 2.

Fix a rational public preference matrix U and a corruption threshold $1 \leq t < n$. Consider a partition of the parties into disjoint sets

$$[n] = Y \cup P_1 \cup \dots \cup P_k, \quad k \geq 2, \quad (16)$$

where every P_ℓ is nonempty and Y may be empty. For each $\ell \in [k]$, let

$$A_\ell = \bigcup_{j \neq \ell} P_j, \quad C_\ell = A_\ell \cup Y = [n] \setminus P_\ell.$$

We require $|C_\ell| \leq t$ for every ℓ . This implies that $n \geq \sum_{\ell=1}^k |P_\ell| \geq k(n-t)$, and thus $t \geq n/2$.

For each $\ell \in [k]$, choose a rational vector

$$\mathbf{v}_\ell \in \mathcal{K}_t(C_\ell) \cap \mathcal{L}_t(A_\ell). \quad (17)$$

For an outcome $a \in [m]$, the a -th entry $v_\ell(a)$ is called the outcome's $\mathbf{v}_\ell$ -score. For a distribution $\mathbf{q} \in \Delta(\Omega)$, its expected score is $\mathbb{E}_{a \leftarrow \mathbf{q}}[v_\ell(a)] = \langle \mathbf{q}, \mathbf{v}_\ell \rangle$. Since $\mathbf{v}_\ell \in \mathcal{V}_m$, the uniform distribution has an expected score of zero.

The condition that $\mathbf{v}_\ell$ belongs to the two sets $\mathcal{K}_t(C_\ell)$ and $\mathcal{L}_t(A_\ell)$ encodes the lone-wolf condition and wolf-minion condition. $\mathbf{v}_\ell \in \mathcal{L}_t(A_\ell)$ means that both $\mathbf{v}_\ell$ and $-\mathbf{v}_\ell$ belong to $\mathcal{K}_t(A_\ell)$. Therefore, in a CSP-fair protocol, a coalition A_ℓ cannot noticeably bias the expected score in either direction. Otherwise, some coalition containing A_ℓ would obtain a noticeable utility gain. $\mathbf{v}_\ell \in \mathcal{K}_t(C_\ell)$ means that a coalition C_ℓ cannot noticeably increase its expected score. Otherwise, some coalition containing C_ℓ would obtain a noticeable utility gain. Although $\mathbf{v}_\ell$ need not be either coalition's utility vector, the proof below shows that violating one of these restrictions contradicts CSP-fairness.

Theorem 6.1. *Fix n, m , a utility matrix U , and $n/2 \leq t < n$. Fix a partition $Y, P_1, \dots, P_k$ of $[n]$ satisfying Equation (16), and score vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ satisfying Equation (17). Define*

$$\eta := \left| \frac{1}{m} \sum_{a \in [m]} \prod_{\ell=1}^k v_\ell(a) \right|. \quad (18)$$

Let $\Pi_\lambda[U]$ be a family of CSP-fair m -sided coin toss protocols that run in a polynomial number of rounds $R(\lambda) \geq 1$. If $\eta > 0$, then there exists a non-uniform PPT rushing semi-malicious adversary $\mathcal{A}$ corrupting a coalition D of size no more than t such that

$$\overline{\text{util}}_D \left(\text{out}_\lambda^{\Pi[U], \mathcal{A}} \right) - \overline{\text{util}}_D \left(\text{out}_\lambda^{\Pi[U], H} \right) \geq \frac{c\eta}{R(\lambda)} - \text{negl}(\lambda) \quad (19)$$

for infinitely many λ and some negligible function $\text{negl}(\cdot)$, where $c > 0$ is some constant independent of λ . Equivalently, a polynomial-round CSP-fair protocol $\Pi[U]$ against coalitions of size at most t exists only if $\eta = 0$ for every such partition and score vector.

Proof. Assume that $\eta > 0$. Without loss of generality, we assume that $|v_\ell(a)| \leq 1$ for every $\ell \in [k]$ and $a \in [m]$. Otherwise, the score vectors can be rescaled, which preserves all conditions above and multiplies η by a fixed positive constant, which can be absorbed into c . We use the same notation for the rescaled vectors and their product average.

Fix an arbitrary security parameter λ and let $\mathbf{p}_H = \text{out}_\lambda^{\Pi[U], H}$. By correctness, $\|\mathbf{p}_H - \boldsymbol{\mu}\|_1 \leq \text{negl}(\lambda)$ for some negligible function $\text{negl}(\cdot)$.

Claim 6.2 (From score bias to utility gain). *Fix $C \subseteq [n]$ and let*

$$\mathbf{v} = \sum_{\substack{D \supseteq C \\ 1 \leq |D| \leq t}} \gamma_D \mathbf{g}_D \in \mathcal{K}_t(C), \quad \gamma_D \geq 0,$$

be nonzero and $\Gamma = \sum_D \gamma_D > 0$. Suppose a non-uniform PPT adversary $\mathcal{A}_C$ controlling C induces an output distribution $\mathbf{q}$ satisfying $\langle \mathbf{q}, \mathbf{v} \rangle \geq \delta$. Then there exists a non-uniform PPT adversary $\mathcal{A}_D$ corrupting some coalition D with $|D| \leq t$ that induces the same output distribution $\mathbf{q}$ and satisfies

$$\overline{\text{util}}_D(\mathbf{q}) - \overline{\text{util}}_D(\mathbf{p}_H) \geq \frac{\delta}{\Gamma} - \text{negl}(\lambda).$$

Proof. Since $\mathbf{v} \in \mathcal{K}_t(C) \subseteq \mathcal{V}_m$, we have $\langle \boldsymbol{\mu}, \mathbf{v} \rangle = 0$. Therefore, by correctness,

$$|\langle \mathbf{p}_H, \mathbf{v} \rangle| = |\langle \mathbf{p}_H - \boldsymbol{\mu}, \mathbf{v} \rangle| \leq \|\mathbf{p}_H - \boldsymbol{\mu}\|_1 \|\mathbf{v}\|_\infty \leq \text{negl}(\lambda).$$

For every D with $\gamma_D > 0$, consider an adversary $\mathcal{A}_D$ corrupting D that internally runs $\mathcal{A}_C$ for the parties in C , while the additional corrupt parties in $D \setminus C$ follow the prescribed protocol honestly. This induces the same output distribution $\mathbf{q}$ as the execution with $\mathcal{A}_C$. By Claim 3.3 and the representation of $\mathbf{v}$,

$$\begin{aligned} \sum_D \gamma_D (\overline{\text{util}}_D(\mathbf{q}) - \overline{\text{util}}_D(\mathbf{p}_H)) &= \sum_D \gamma_D \langle \mathbf{q} - \mathbf{p}_H, \mathbf{g}_D \rangle \\ &= \langle \mathbf{q} - \mathbf{p}_H, \mathbf{v} \rangle \\ &= \langle \mathbf{q}, \mathbf{v} \rangle - \langle \mathbf{p}_H, \mathbf{v} \rangle \\ &\geq \delta - \text{negl}(\lambda). \end{aligned}$$

Since all γ_D are nonnegative and $\Gamma = \sum_D \gamma_D > 0$, there must be some D with $\gamma_D > 0$ such that

$$\overline{\text{util}}_D(\mathbf{q}) - \overline{\text{util}}_D(\mathbf{p}_H) \geq \frac{\delta}{\Gamma} - \text{negl}(\lambda).$$

□

For every $\ell \in [k]$, fix conic representations witnessing $\mathbf{v}_\ell \in \mathcal{L}_t(A_\ell)$ and $\mathbf{v}_\ell \in \mathcal{K}_t(C_\ell)$. Since $\mathbf{v}_\ell \in \mathcal{L}_t(A_\ell)$, both $\mathbf{v}_\ell$ and $-\mathbf{v}_\ell$ belong to $\mathcal{K}_t(A_\ell)$. Applying Claim 6.2 to these fixed representations, we obtain a constant $c_0 > 0$ such that an adversary controlling A_ℓ whose expected $\mathbf{v}_\ell$ -score has absolute value at least δ yields some admissible coalition a utility gain of at least $c_0\delta - \text{negl}(\lambda)$. Similarly, an adversary controlling C_ℓ whose expected $\mathbf{v}_\ell$ -score is at least δ yields some admissible coalition a utility gain of at least $c_0\delta - \text{negl}(\lambda)$.

Abort Executions. Following Cleve's construction [Cle86] of a sequence of fail-stop adversaries who abort before or after a round, we consider the following sequence of random variables $Z_{\ell,r}$ for $\ell \in [k]$ and $r \in \{0, \dots, R\}$.

Sample an independent honest random tape rt_i for every party i , and use the same tuple $\text{rt} = (\text{rt}_1, \dots, \text{rt}_n)$ in all executions below. For every $\ell \in [k]$ and each round $r \in \{0, \dots, R\}$, consider the execution in which all parties follow the prescribed protocol through round r , after which all parties in A_ℓ abort before sending their round- $(r+1)$ messages. Let $Z_{\ell,r}$ denote the $\mathbf{v}_\ell$ -score of the resulting outcome. For every fixed realization of rt , the variables $Z_{\ell,r}$ are deterministic.

For a fixed ℓ , the sequence $Z_{\ell,0}, \dots, Z_{\ell,R}$ is the analogue of Cleve's abort sequence: $Z_{\ell,r-1}$ and $Z_{\ell,r}$ correspond to aborting immediately before and immediately after round r , respectively. Because the executions use the same random tapes, their difference isolates the effect of allowing the round- r messages to be sent before A_ℓ aborts. Here, instead of tracking the bias of a binary outcome, we track the $\mathbf{v}_\ell$ -score of an m -sided outcome, and we consider k such sequences.

$Z_{\ell,0}$ is the score when A_ℓ aborts immediately before sending any message, whereas $Z_{\ell,R}$ is the score of the honest outcome. When $r = R$, the abort occurs only after all protocol rounds have completed, so the execution is the honest execution with the chosen randomness $\mathbf{rt}$. Hence, there is a single honest outcome $a_H \in [m]$ and $Z_{\ell,R} = v_\ell(a_H)$ for every $\ell \in [k]$. Define

$$\Phi_r := \mathbb{E}_{\mathbf{rt}} \left[\prod_{\ell=1}^k Z_{\ell,r} \right].$$

Recall that in an honest execution, the distribution of the outcome is $\mathbf{p}_H = (p_H(1), \dots, p_H(m))$. We have

$$\Phi_R = \mathbb{E}_{\mathbf{rt}} \left[\prod_{\ell=1}^k Z_{\ell,R} \right] = \sum_{a \in [m]} p_H(a) \prod_{\ell=1}^k v_\ell(a).$$

Since $|v_\ell(a)| \leq 1$ and $\|\mathbf{p}_H - \boldsymbol{\mu}\|_1 \leq \text{negl}(\lambda)$, we have

$$\left| \Phi_R - \frac{1}{m} \sum_{a \in [m]} \prod_{\ell=1}^k v_\ell(a) \right| \leq \|\mathbf{p}_H - \boldsymbol{\mu}\|_1 \leq \text{negl}(\lambda).$$

Since $\eta > 0$ is fixed, for all sufficiently large λ ,

$$|\Phi_R| \geq \frac{3\eta}{4}. \tag{20}$$

We split the remaining analysis into two cases.

Case 1: if $|\Phi_0| \geq \frac{\eta}{4}$. Let $\mathbf{rt}_Y$ be the tuple of random tapes of the parties in Y , and define

$$z_\ell(y) := \mathbb{E}[Z_{\ell,0} \mid \mathbf{rt}_Y = y].$$

After A_ℓ aborts immediately, only the parties in $P_\ell \cup Y$ remain. Conditioned on $\mathbf{rt}_Y$, the random variable $Z_{\ell,0}$ depends only on the random tapes of the parties in P_ℓ . Therefore, $Z_{1,0}, \dots, Z_{k,0}$ are conditionally independent given $\mathbf{rt}_Y$, and

$$\Phi_0 = \mathbb{E}_{\mathbf{rt}_Y} \left[\prod_{\ell=1}^k z_\ell(\mathbf{rt}_Y) \right].$$

Since $|z_\ell(y)| \leq 1$ for every ℓ and y , for every fixed y we have $\left| \prod_{\ell=1}^k z_\ell(y) \right| \leq |z_1(y)|$. Therefore,

$$\frac{\eta}{4} \leq |\Phi_0| \leq \mathbb{E} \left[\left| \prod_{\ell=1}^k z_\ell(\mathbf{rt}_Y) \right| \right] \leq \mathbb{E}[|z_1(\mathbf{rt}_Y)|].$$

If $|\mathbb{E}Z_{1,0}| = |\mathbb{E}z_1(\mathbf{rt}_Y)| \geq \frac{\eta}{8}$, then the adversary controlling A_1 that aborts immediately has expected score at least $\eta/8$ in direction either $\mathbf{v}_1$ or $-\mathbf{v}_1$. Since both directions belong to $\mathcal{K}_t(A_1)$, by Claim 6.2,

there is an adversary corrupting an admissible coalition with a utility gain of at least $c_0\eta/8 - \text{negl}(\lambda)$ compared to the honest case.

Otherwise, we have $|\mathbb{E}z_1(\mathbf{rt}_Y)| < \eta/8$. Then,

$$\mathbb{E}[\max\{z_1(\mathbf{rt}_Y), 0\}] = \frac{\mathbb{E}|z_1(\mathbf{rt}_Y)| + \mathbb{E}z_1(\mathbf{rt}_Y)}{2} \geq \frac{\eta}{16}.$$

Hence there exists a random tape $\mathbf{rt}_Y^*$ for the parties in Y such that $z_1(\mathbf{rt}_Y^*) \geq \frac{\eta}{16}$. Consider an adversary controlling $C_1 = A_1 \cup Y$ that makes A_1 abort immediately and makes the parties in Y follow the prescribed protocol using the fixed tape $\mathbf{rt}_Y^*$. The remaining parties in P_1 are honest, so this adversary has expected $\mathbf{v}_1$ -score exactly $z_1(\mathbf{rt}_Y^*)$. The tape $\mathbf{rt}_Y^*$ has polynomial length and can be included in the adversary's non-uniform advice. Since $\mathbf{v}_1 \in \mathcal{K}_t(C_1)$, Claim 6.2 gives an adversary corrupting an admissible coalition with a utility gain of at least $c_0\eta/16 - \text{negl}(\lambda)$ compared to the honest case. If $Y = \emptyset$, then $z_1(\mathbf{rt}_Y)$ is constant, so this second subcase cannot occur.

Thus, if $|\Phi_0| \geq \eta/4$, there is an adversary corrupting an admissible coalition with a noticeable utility gain compared to the honest case.

Case 2: if $|\Phi_0| < \frac{\eta}{4}$. Together with (20), we have $|\Phi_R - \Phi_0| \geq \frac{\eta}{2}$. Thus, there must exist a round $r^* \in [R]$ such that

$$|\Phi_{r^*} - \Phi_{r^*-1}| \geq \frac{\eta}{2R}. \quad (21)$$

Define

$$\Lambda_{\ell, r^*} := \prod_{j < \ell} Z_{j, r^*} \prod_{j > \ell} Z_{j, r^*-1}.$$

For every fixed choice of the parties' random tapes $\mathbf{rt}$,

$$\prod_{\ell=1}^k Z_{\ell, r^*} - \prod_{\ell=1}^k Z_{\ell, r^*-1} = \sum_{\ell=1}^k \Lambda_{\ell, r^*} (Z_{\ell, r^*} - Z_{\ell, r^*-1}).$$

Taking expectations and using (21), we obtain an index $\ell \in [k]$ and a sign $\sigma \in \{-1, 1\}$ such that

$$\sigma \mathbb{E}[\Lambda_{\ell, r^*} (Z_{\ell, r^*} - Z_{\ell, r^*-1})] \geq \frac{\eta}{2kR}. \quad (22)$$

We now use this correlation to construct a rushing adversary. First consider an adversary $\mathcal{B}$ controlling A_ℓ that follows the prescribed protocol through round $r^* - 1$, using independent honest random tapes. After receiving the honest parties' round- r^* messages, it flips an independent fair coin. With probability $1/2$, the parties in A_ℓ abort without sending their round- r^* messages; otherwise, they send their prescribed round- r^* messages and then immediately abort. Its expected $\mathbf{v}_\ell$ -score is

$$z_{\text{base}} := \frac{1}{2} \mathbb{E}[Z_{\ell, r^*-1} + Z_{\ell, r^*}]. \quad (23)$$

Next consider another adversary $\mathcal{B}'$ controlling $C_\ell = A_\ell \cup Y$. The parties in Y follow the prescribed protocol using independent honest random tapes, while the parties in A_ℓ behave as above except that their abort probability depends on Λ_{ℓ, r^*} .

In round r^* , after receiving the honest parties' messages from P_ℓ , the adversary $\mathcal{B}'$ computes the scores needed for Λ_{ℓ, r^*} . For every $j \neq \ell$, all parties in $P_j \cup Y$ belong to C_ℓ , so $\mathcal{B}'$ knows their chosen randomness and their states through round $r^* - 1$. To compute Z_{j, r^*-1} , it simulates these parties from those states with A_j aborting before round r^* . To compute Z_{j, r^*} , it first supplies their prescribed round- r^* incoming messages and then simulates the continuation with A_j aborting before round $r^* + 1$. The required round- r^* messages from P_ℓ have already been received, while the

messages from parties in C_ℓ can be computed from their known states and randomness. No private message whose two endpoints are both in P_ℓ is needed, because every party in $P_\ell \subseteq A_j$ is absent from the simulated continuation. Thus $\mathcal{B}'$ computes every factor of Λ_{ℓ,r^*} in polynomial time before deciding whether A_ℓ sends its round- r^* messages.

The adversary lets the parties in A_ℓ send their prescribed round- r^* messages with probability $\xi = \frac{1+\sigma\Lambda_{\ell,r^*}}{2} \in [0, 1]$, and otherwise abort without sending. Since the score vectors are fixed rational vectors, ξ is a rational number and can be computed efficiently up to negligible statistical error. Let z_{rush} denote its expected $\mathbf{v}_\ell$ -score. Then we have

$$\begin{aligned} z_{\text{rush}} - z_{\text{base}} &= \mathbb{E}[\xi Z_{\ell,r^*} + (1 - \xi)Z_{\ell,r^*-1}] \\ &\quad - \frac{1}{2}\mathbb{E}[Z_{\ell,r^*} + Z_{\ell,r^*-1}] \\ &= \frac{\sigma}{2}\mathbb{E}[\Lambda_{\ell,r^*}(Z_{\ell,r^*} - Z_{\ell,r^*-1})] \geq \frac{\eta}{4kR} \quad (\text{by (22)}). \end{aligned}$$

If

$$z_{\text{base}} < -\frac{\eta}{8kR},$$

then the baseline A_ℓ -adversary has expected score greater than $\eta/(8kR)$ in direction $-\mathbf{v}_\ell$. Since $-\mathbf{v}_\ell \in \mathcal{K}_t(A_\ell)$, Claim 6.2 gives an admissible coalition a utility gain of at least $\frac{c_0\eta}{8kR} - \text{negl}(\lambda)$.

Otherwise, we have

$$z_{\text{rush}} \geq -\frac{\eta}{8kR} + \frac{\eta}{4kR} = \frac{\eta}{8kR},$$

up to the negligible sampling error. Since $\mathbf{v}_\ell \in \mathcal{K}_t(C_\ell)$, applying Claim 6.2 again gives an admissible coalition the same lower bound on its utility gain.

Combining the two cases shows that, for every sufficiently large λ , one of the adversaries constructed above controls some A_ℓ or C_ℓ and achieves a score bias. By Claim 6.2, this yields an adversary controlling some coalition D with $|D| \leq t$ whose utility gain is at least $\frac{c_0\eta}{8kR(\lambda)} - \text{negl}(\lambda)$. In Case 1 the utility gain is in fact constant, and hence also satisfies this bound since $R(\lambda) \geq 1$.

The coalition D returned by Claim 6.2 may depend on λ . However, since there are only finitely many coalitions of size at most t , there is a fixed D for which the above bound holds for infinitely many λ .

Fix such a coalition D . We construct a single non-uniform PPT adversary $\mathcal{A}_D$ controlling D . For every security parameter in this infinite set, $\mathcal{A}_D$ receives advice specifying the corresponding attack: abort at the beginning or in round r^* , the index ℓ , the sign σ , and the fixed tape rt_Y^* when relevant. The adversary first runs the corresponding adversary controlling A_ℓ or C_ℓ , and then extends it to coalition D exactly as in Claim 6.2, with the remaining corrupt parties acting honestly. For all other security parameters, $\mathcal{A}_D$ follows the prescribed protocol.

Thus, for infinitely many λ ,

$$\overline{\text{util}}_D \left(\text{out}_\lambda^{\Pi[U], \mathcal{A}_D} \right) - \overline{\text{util}}_D \left(\text{out}_\lambda^{\Pi[U], H} \right) \geq \frac{c_0\eta}{8kR(\lambda)} - \text{negl}(\lambda).$$

Taking $c = c_0/(8k)$ proves (19). Since $\eta > 0$ is fixed and $R(\lambda)$ is polynomially bounded, $c\eta/R(\lambda)$ is non-negligible, contradicting CSP-fairness against coalitions of size at most t . $\square$

References

- [AO16] Bar Alon and Eran Omri. Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In *Theory of Cryptography Conference (TCC 2016-B), Part I*, pages 307–335. Springer, 2016.

- [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 (SODA)*, 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.
- [Blu83] Manuel Blum. Coin flipping by telephone: A protocol for solving impossible problems. *SIGACT News*, 15(1):23–27, January 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 *Advances in Cryptology – CRYPTO 2021, Part II*, pages 3–32. Springer, 2021.
- [CD17] Ignacio Cascudo and Bernardo David. SCRAPE: Scalable randomness attested by public entities. In *Applied Cryptography and Network Security*, volume 10355 of *Lecture Notes in Computer Science*, pages 537–556. Springer, 2017.
- [CGL⁺18] Kai-Min Chung, Yue Guo, Wei-Kai Lin, Rafael Pass, and Elaine Shi. Game theoretic notions of fairness in multi-party coin toss. In *Theory of Cryptography Conference (TCC 2018), Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 563–596. Springer, 2018.
- [CKS05] Christian Cachin, Klaus Kursawe, and Victor Shoup. Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography. *Journal of Cryptology*, 18(3):219–246, 2005.
- [Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 364–369. ACM, 1986.
- [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 (TCC)*, pages 450–467. Springer, 2011.
- [Fei99] Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 142–153. IEEE, 1999.
- [GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
- [GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229. ACM, 1987.

- [HT14] Iftach Haitner and Eliad Tsfasia. An almost-optimally fair three-party coin-flipping protocol. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 408–416. ACM, 2014.
- [KKZ25] Ohad Klein, Ilan Komargodski, and Chenzhi Zhu. On the (im)possibility of game-theoretically fair leader election protocols. In *Theory of Cryptography Conference (TCC 2024), Part II*, pages 383–412. Springer, 2025.
- [KMSW22] Ilan Komargodski, Shin’ichiro Matsuo, Elaine Shi, and Ke Wu. $\log^*$ -round game-theoretically fair leader election. In *Advances in Cryptology – CRYPTO 2022, Part III*, pages 409–438. Springer, 2022.
- [Lin03] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology*, 16(3):143–184, 2003.
- [MNS16] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. *Journal of Cryptology*, 29(3):491–513, 2016.
- [Rab83] Michael O. Rabin. Randomized Byzantine generals. In *24th Annual Symposium on Foundations of Computer Science*, pages 403–409. IEEE, 1983.
- [RBO89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (STOC)*, pages 73–85. ACM, 1989.
- [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.
- [Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.
- [SJKK⁺17] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J. Fischer, and Bryan Ford. Scalable bias-resistant distributed randomness. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 444–460, 2017.
- [TSW24] Sri AravindaKrishnan Thyagarajan, Pratik Soni, and Ke Wu. Game-theoretically fair distributed sampling. In *Advances in Cryptology – CRYPTO 2024, Part VIII*, pages 207–239. Springer, 2024.
- [WAS22] Ke Wu, Gilad Asharov, and Elaine Shi. A complete characterization of game-theoretically fair, multi-party coin toss. In *Advances in Cryptology – EUROCRYPT 2022, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 120–149. Springer, 2022.
- [Yao82] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 160–164. IEEE, 1982.

- [ZW25] Forest Zhang and Ke Wu. Game-theoretically fair coin toss with arbitrary preferences. In *Advances in Cryptology – ASIACRYPT 2025, Part VIII*, volume 16252 of *Lecture Notes in Computer Science*, pages 549–581. Springer, 2025.

Appendix

A Computing the Public Protocol Parameters

This appendix gives explicit polynomial-time procedures for the public parameters used in Section 4.2. We first compute an abort-safe opening order, then construct the punishment directions, and finally choose a common walk bias α . Proposition A.6 combines their guarantees. Polynomial time and description length are measured in the binary input length of (n, m, t, U) .

A.1 Opening Orders

Recall the centered utility matrix $G = (\mathbf{g}_1 \cdots \mathbf{g}_n)$ from Section 5, so $G\mathbf{x} = \sum_{i=1}^n x_i \mathbf{g}_i$ for $\mathbf{x} \in \mathbb{R}^n$. The following algorithm gives the opening order when $\text{rank } G \geq n-1$, or equivalently $\dim \ker G \leq 1$.

Algorithm 1 Rank-condition opening-order rule

Input: a rational instance (n, m, t, U) with $1 \leq t < n$ and $\text{rank } G \geq n-1$.

Output: a rule for evaluating π_S for $S = [n] \setminus A$ and every abort set A with $|A| \leq t$.

- 1: Form the centered utility map G and compute a basis of $\ker G$ by Gaussian elimination.
 - 2: Save the index order $1, \dots, n$.
 - 3: **if** $\dim \ker G = 1$ **then**
 - 4: Choose a nonzero generator $\boldsymbol{\kappa}$ of $\ker G$.
 - 5: **if** one sign of $\boldsymbol{\kappa}$ is nonnegative and its entries are not all equal **then**
 - 6: Choose the nonnegative sign and save the order of decreasing κ_i , breaking ties by index.
 - 7: **return** the rule: π_S is the saved order restricted to $S = [n] \setminus A$.
-

Algorithm 1 implements the order constructed in the proof of Theorem 5.1, so its correctness follows from that theorem. Gaussian elimination and sorting take polynomial time; restricting the saved order after an abort takes $O(n)$ time.

Without the rank condition, Algorithm 2 tests whether an abort-safe opening order exists and finds one when it does. It uses exact rational linear-programming (LP) feasibility tests.

The next lemma explains why it is enough to find one order and delete parties as they abort. A suffix of an order consists of some party and all parties after it.

Lemma A.1. *U satisfies the abort-safe opening-order condition at threshold t if and only if the parties can be ordered by a permutation π of $[n]$ such that*

$$\mathcal{K}_t(F) \cap \mathcal{L}_t(\{i\}) = \{\mathbf{0}\} \quad \text{for every nonempty suffix } F \text{ of } \pi \text{ and every } i \in F. \quad (24)$$

For every A with $|A| \leq t$, deleting the parties in A from π gives an order satisfying Definition 4.5 for that A .

Proof. Suppose the condition holds, and take its order π for $A = \emptyset$. If (24) failed, some nonzero $\mathbf{y} \in \mathcal{K}_t(F) \cap \mathcal{L}_t(\{i\})$ could be written as $\mathbf{y} = \sum_{F \subseteq \mathcal{C}, 1 \leq |C| \leq t} \theta_C \mathbf{g}_C$ with $\theta_C \geq 0$. For any term with $\theta_C > 0$, adding all the other terms to $-\mathbf{y}$ shows that $-\theta_C \mathbf{g}_C \in \mathcal{K}_t(\{i\})$. Hence every nonzero $\mathbf{g}_C$ used in the sum lies in $\mathcal{L}_t(\{i\})$. At least one is nonzero because $\mathbf{y} \neq \mathbf{0}$; choose it. Because $\mathcal{C}$ contains the suffix F , no party outside $\mathcal{C}$ occurs after i , contradicting (12).

Conversely, suppose π satisfies (24). Fix A with $|A| \leq t$ and delete A from π . If (12) failed for some $\mathcal{C}$ and i , every remaining party after i would belong to $\mathcal{C}$. The deleted parties also belong to $\mathcal{C}$ because $A \subseteq \mathcal{C}$. Thus the suffix F of π beginning at i is contained in $\mathcal{C}$, so $\mathbf{g}_C \in \mathcal{K}_t(F)$. Every coalition containing $A \cup \{i\}$ also contains i , and therefore $\mathcal{L}_t(A \cup \{i\}) \subseteq \mathcal{L}_t(\{i\})$. The failed implication now puts the nonzero vector $\mathbf{g}_C$ in both sets in (24), a contradiction. $\square$

Algorithm 2 builds the opening order from last to first. Suppose that the parties in F have already been placed at the end and that $\mathcal{K}_t(F) \cap \mathcal{L}_t(\{i\}) = \{\mathbf{0}\}$ for every $i \in F$. To place $v \notin F$ immediately before them, it is enough to check

$$\mathcal{K}_t(F \cup \{v\}) \cap \mathcal{L}_t(\{v\}) = \{\mathbf{0}\}. \quad (25)$$

Indeed, this is the condition for v , while the condition for every $i \in F$ remains true because

$$\mathcal{K}_t(F \cup \{v\}) \subseteq \mathcal{K}_t(F).$$

To check (25), put $W = F \cup \{v\}$. For every nonempty $W \subseteq [n]$, define

$$\begin{aligned} \mathcal{Q}_t(W) &:= \left\{ (\mathbf{x}, s) \in \mathbb{R}^n \times \mathbb{R} : \begin{array}{ll} s \geq 0, & 0 \leq x_j \leq s \quad (j \in [n]), \\ x_i = s & (i \in W), \quad \sum_{j=1}^n x_j \leq ts \end{array} \right\}, \\ \mathcal{K}_t(W) &= \{G\mathbf{x} : (\mathbf{x}, s) \in \mathcal{Q}_t(W) \text{ for some } s\}. \end{aligned} \quad (26)$$

The equality for $\mathcal{K}_t(W)$ follows by taking $\mathbf{x}$ to be a nonnegative combination of coalition incidence vectors and s to be the sum of their coefficients. Conversely, when $s > 0$, the vector $\mathbf{x}/s$ lies in the polytope defined by $0 \leq x_j/s \leq 1$, $x_i/s = 1$ for $i \in W$, and $\sum_j x_j/s \leq t$. Its vertices are 0–1 vectors: two fractional coordinates could be changed in opposite directions, while one fractional coordinate could be changed unless the last inequality is tight, in which case the integer t forces that coordinate to be integral. Thus $\mathbf{x}/s$ is a convex combination of incidence vectors of coalitions that contain W and have size at most t . When $s = 0$, the constraints force $\mathbf{x} = \mathbf{0}$.

Since $\mathcal{K}_t(W) \subseteq \mathcal{K}_t(\{v\})$, (25) holds exactly when all the following systems, one for each $j \in [m]$ and $\sigma \in \{-1, 1\}$, are infeasible:

$$\begin{aligned} (\mathbf{x}, s) &\in \mathcal{Q}_t(W), & (\mathbf{z}, r) &\in \mathcal{Q}_t(\{v\}), \\ G\mathbf{x} + G\mathbf{z} &= \mathbf{0}, & \sigma(G\mathbf{x})_j &\geq 1. \end{aligned} \quad (27)$$

If a system is feasible, $G\mathbf{x}$ is a nonzero vector in the intersection in (25). Conversely, any nonzero vector in that intersection has a nonzero coordinate; choose its sign and scale both cone representations until the last inequality holds. If $|W| > t$, then $\mathcal{K}_t(W) = \{\mathbf{0}\}$, so no LP is needed.

Algorithm 2 Opening order without the rank condition

Input: a rational instance (n, m, t, U) with $1 \leq t < n$.

Output: an opening order if one exists; otherwise, report failure.

- 1: Form the centered utility map G .
 - 2: $F \leftarrow \emptyset$ and $R \leftarrow []$.
 - 3: **while** $|F| < n$ **do**
 - 4: Examine $v \notin F$ in index order and choose the first one for which all systems in (27) are infeasible.
 - 5: **if** no such v exists **then**
 - 6: **return** “the opening-order condition fails at threshold t ”.
 - 7: Append v to R and set $F \leftarrow F \cup \{v\}$.
 - 8: **return** $\pi = \text{reverse}(R)$ and, for $S = [n] \setminus A$, the rule $\pi_S = \pi|_S$.
-

Lemma A.2 (Computing an abort-safe opening order). *Algorithm 2 decides in polynomial time whether the abort-safe opening-order condition holds. On success, it returns a single order whose restriction works for every abort set of size at most t . It makes $O(mn^2)$ LP feasibility tests, each with $2n + 2$ variables and $O(n + m)$ constraints.*

Proof. The nonempty sets F produced by the algorithm are the suffixes of the returned order. By construction they satisfy (24), so Lemma A.1 shows that the returned order works.

To see why the backward search cannot miss an order, suppose a permutation π^* satisfies (24). Among the parties not yet in F , choose the one occurring latest in π^* , and call it v . If F^* is the suffix of π^* beginning at v , every other party in F^* already belongs to F . Hence $F^* \subseteq F \cup \{v\}$ and

$$\mathcal{K}_t(F \cup \{v\}) \subseteq \mathcal{K}_t(F^*).$$

Applying (24) to F^* shows that v passes the test. Thus the algorithm cannot get stuck when an order exists, and a failure report is correct.

There are at most n stages and n candidates per stage, with $2m$ LP tests per candidate. Each test has $2n + 2$ variables, $O(n + m)$ rational constraints, and coefficients having polynomially many bits. Exact rational linear programming is polynomial-time solvable [GLS88], giving the stated bound. Restricting the stored order after an abort takes $O(n)$ time. $\square$

A.2 Punishment Directions

A vector $\mathbf{z} \in \mathcal{V}_m$ preserves the sum of the probabilities. As in Section 3.2, it is a punishment direction for A if $\langle \mathbf{z}, \mathbf{g}_C \rangle \leq 0$ for every coalition $C \supseteq A$ with $1 \leq |C| \leq t$. The algorithm finds enough linearly independent punishment directions to span all of them. It then clears their denominators, adds them, and makes the result small enough to use as an update. The resulting vector does not increase any such coalition's utility, and strictly decreases it whenever $\mathbf{g}_C \notin \mathcal{L}_t(A)$.

We use an integer-scaled copy of the utilities. Let d be m times the product of the positive denominators of the entries of U , each written in lowest terms, and put $\tilde{U} = dU$ and $\tilde{\mathbf{g}}_i = d\mathbf{g}_i$. The vectors $\tilde{\mathbf{g}}_i$ have integer entries; write $\tilde{g}_i(j)$ for the entry at outcome $j \in \Omega$. For every coalition C , write $\tilde{\mathbf{g}}_C = \sum_{i \in C} \tilde{\mathbf{g}}_i = d\mathbf{g}_C$. This common positive scaling preserves utility comparisons, the cones $\mathcal{K}_t(A)$, their lineality spaces, and $\ker G$. The original U and $\mathbf{g}_i$ keep their meanings from the model. The scaled copy is used only to compute the public protocol parameters.

Set

$$H := n + \sum_{i,j} |\tilde{g}_i(j)|, \quad \Delta := m!H^m, \quad B := 2m^3t\Delta, \quad \gamma := 1/B. \quad (28)$$

These constants depend only on the public instance.

Fix a nonempty abort set A with $|A| \leq t$. Define the bounded polytope of punishment directions

$$\mathcal{P}_A = \left\{ \mathbf{z} \in \mathcal{V}_m : \begin{array}{l} -1 \leq z_j \leq 1 \text{ for every } j \in \Omega, \\ \langle \mathbf{z}, \tilde{\mathbf{g}}_C \rangle \leq 0 \text{ for every } C \supseteq A \text{ with } |C| \leq t \end{array} \right\}. \quad (29)$$

Membership in $\mathcal{V}_m$ says that moving in direction $\mathbf{z}$ preserves total probability. The box constraints keep the search bounded. The utility constraints say exactly that the shift cannot help any coalition that could be responsible for abort set A .

For $\mathbf{w} \in \mathbb{R}^m$, let $\text{LP}_A(\mathbf{w})$ denote the following deterministic LP subroutine: maximize $\langle \mathbf{w}, \mathbf{z} \rangle$ over $\mathbf{z} \in \mathcal{P}_A$; among all maximizers, minimize z_1 , then z_2 , and so on. This tie-breaking rule selects one fixed vertex of $\mathcal{P}_A$.

The subroutine checks the zero-sum and box constraints directly. It does not need to list every coalition constraint. Given a candidate $\mathbf{z}$, compute the individual utility changes $s_i = \langle \mathbf{z}, \tilde{\mathbf{g}}_i \rangle$ for $i \notin A$. Starting from A , add up to $t - |A|$ parties having the largest positive values of s_i . The resulting coalition maximizes $\langle \mathbf{z}, \tilde{\mathbf{g}}_C \rangle$ among all admissible $C \supseteq A$. Therefore a positive value gives a violated constraint, while a nonpositive value certifies that all coalition constraints hold. This is the separation oracle used by the LP solver.

Algorithm 3 Online punishment update

Input: the integral centered vectors $\tilde{\mathbf{g}}_i$ and B defined above, and a nonempty cumulative abort set $A \subseteq [n]$ with $|A| \leq t$.

Output: a punishment direction $\mathbf{a}_A$ for use in $\text{dist-update}(\mathbf{p}, A) = \mathbf{p} + \mathbf{a}_A$ when an opening fails.

```
1:  $\mathcal{B} \leftarrow []$ 
2: loop
3:   Compute a basis  $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$  of  $\text{span}(\mathcal{B})^\perp$  by Gaussian elimination with fixed pivot order.
4:    $\text{new} \leftarrow \text{none}$ 
5:   for  $\mathbf{w}$  in  $(\mathbf{w}_1, -\mathbf{w}_1, \dots, \mathbf{w}_\ell, -\mathbf{w}_\ell)$  do
6:      $\mathbf{z} \leftarrow \text{LP}_A(\mathbf{w})$ 
7:     if  $\langle \mathbf{w}, \mathbf{z} \rangle > 0$  then
8:        $\text{new} \leftarrow \mathbf{z}$ 
9:       break
10:  if  $\text{new} = \text{none}$  then
11:    break
12:  Append  $\text{new}$  to  $\mathcal{B}$ 
13:  $\mathbf{z}_A \leftarrow \mathbf{0}$ 
14: for each  $\mathbf{v} \in \mathcal{B}$  do
15:   Let  $d_{\mathbf{v}}$  be the least common multiple of the denominators of the coordinates of  $\mathbf{v}$ 
16:    $\hat{\mathbf{v}} \leftarrow d_{\mathbf{v}} \mathbf{v}$ 
17:    $\mathbf{z}_A \leftarrow \mathbf{z}_A + \hat{\mathbf{v}}$ 
18:  $\mathbf{a}_A \leftarrow \mathbf{z}_A / B$ 
19: return  $\mathbf{a}_A$ 
```

Lemma A.3 (Spanning and strict punishment). *Let $\mathcal{B} = (\mathbf{v}_1, \dots, \mathbf{v}_r)$ and $\mathbf{z}_A$ be the list and integer sum constructed by Algorithm 3. The loop terminates after at most m appends, and $\text{span}(\mathcal{B}) = \text{span}(\mathcal{P}_A)$. For every coalition $\mathcal{C} \supseteq A$ with $|\mathcal{C}| \leq t$,*

$$\langle \mathbf{z}_A, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle = \begin{cases} 0, & \mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(A), \\ \leq -1, & \mathbf{g}_{\mathcal{C}} \notin \mathcal{L}_t(A). \end{cases} \quad (30)$$

Proof. Whenever the algorithm appends a vector $\mathbf{v}$, it has found a probe $\mathbf{w}$ that is orthogonal to all previously selected vectors but satisfies $\langle \mathbf{w}, \mathbf{v} \rangle > 0$. Therefore $\mathbf{v}$ is not in the span of the earlier vectors. Each append increases the dimension of the span by one, so there can be at most m appends.

Now consider the final pass, when no new vector is found. Let $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$ be the basis of $\text{span}(\mathcal{B})^\perp$ used on that pass. The zero vector lies in $\mathcal{P}_A$, so the optimum of every LP is at least zero. Since neither $\mathbf{w}_k$ nor $-\mathbf{w}_k$ gives a positive optimum, both optima are exactly zero. Hence every $\mathbf{z} \in \mathcal{P}_A$ satisfies $\langle \mathbf{w}_k, \mathbf{z} \rangle = 0$ for every k , so it lies in $\text{span}(\mathcal{B})$. Every punishment direction can be multiplied by a sufficiently small positive number to put it in $\mathcal{P}_A$. Thus the same span contains every punishment direction.

Every selected $\mathbf{v}_q$ belongs to $\mathcal{P}_A$. Thus, for every $\mathcal{C} \supseteq A$ with $|\mathcal{C}| \leq t$, $\langle \mathbf{v}_q, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle \leq 0$.

First suppose $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(A)$. By definition of the lineality space, both $\tilde{\mathbf{g}}_{\mathcal{C}}$ and $-\tilde{\mathbf{g}}_{\mathcal{C}}$ lie in $\mathcal{K}_t(A)$. A punishment direction has nonpositive inner product with both of them. The two inequalities force $\langle \mathbf{v}_q, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle = 0$ for every selected direction.

Next suppose $\mathbf{g}_{\mathcal{C}} \notin \mathcal{L}_t(A)$. By Lemma 4.3, there is a punishment direction $\mathbf{b}$ with $\langle \mathbf{b}, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle < 0$ and nonpositive inner product with every vector in $\mathcal{K}_t(A)$. Multiplying $\mathbf{b}$ by a sufficiently small positive number puts it inside the box $[-1, 1]^m$, so the scaled vector belongs to $\mathcal{P}_A$. The spanning property just proved puts it in $\text{span}(\mathcal{B})$. If every selected direction had zero inner product with $\tilde{\mathbf{g}}_{\mathcal{C}}$, then every vector in their span would also have zero inner product with $\tilde{\mathbf{g}}_{\mathcal{C}}$, contradicting the

strict inequality for the scaled $\mathbf{b}$. Therefore at least one selected direction has strictly negative inner product with $\tilde{\mathbf{g}}_{\mathcal{C}}$.

For each selected vector, the algorithm multiplies by a positive integer to clear denominators. Because every $\tilde{\mathbf{g}}_{\mathcal{C}}$ is integral after the setup, $\langle \hat{\mathbf{v}}_q, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle$ is an integer. All these integers are nonpositive, and for a punishable coalition at least one is strictly negative. This proves (30). $\square$

Lemma A.4 (Bounded efficient punishment updates). *For every nonempty A with $|A| \leq t$, Algorithm 3 computes in polynomial time a vector $\mathbf{a}_A$ of polynomial description length satisfying*

$$\begin{aligned} \langle \mathbf{a}_A, \mathbf{1} \rangle &= 0, & \|\mathbf{a}_A\|_1 &\leq \frac{1}{2mt}, \\ \langle \mathbf{a}_A, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle &= 0 & \text{if } \mathbf{g}_{\mathcal{C}} \in \mathcal{L}_t(A), & \quad (A \subseteq \mathcal{C}, |\mathcal{C}| \leq t). \\ \langle \mathbf{a}_A, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle &\leq -\gamma & \text{if } \mathbf{g}_{\mathcal{C}} \notin \mathcal{L}_t(A), & \end{aligned} \quad (31)$$

For the original utilities, the same statements hold with $\tilde{\mathbf{g}}_{\mathcal{C}}$ replaced by $\mathbf{g}_{\mathcal{C}}$ and γ by γ/d . In particular, $\mathbf{a}_A$ satisfies Lemma 4.3.

Proof. By Lemma A.3, division by B gives the utility inequalities with margin $\gamma = 1/B$. It remains to bound the magnitude and description length of the returned update.

The deterministic tie-breaking in LP_A chooses a vertex of the polytope $\mathcal{P}_A$. To see this, first restrict to the face on which the main objective is optimal. Successively minimizing $z_1, z_2, \dots, z_m$ moves to smaller faces until one point remains; that point is a vertex of the original polytope.

At a vertex in $\mathbb{R}^m$, enough tight constraints meet to pin down all m coordinates. Equivalently, a vertex is determined by m linearly independent active constraints. After the utility scaling, all coefficients and right-hand sides in these constraints are integers of absolute value at most H . Cramer's rule therefore bounds a common denominator and all coordinate numerators of every selected LP vertex by $\Delta = m!H^m$ [Sch86]. The least common multiple of the coordinate denominators divides this common denominator. Clearing denominators of a selected vector therefore produces an integer vector whose coordinates have absolute value at most Δ . By Lemma A.3, at most m vectors are selected, so

$$|z_A(j)| \leq m\Delta \quad (j \in \Omega), \quad \|\mathbf{z}_A\|_1 \leq m^2\Delta.$$

Using $B = 2m^3t\Delta$, we obtain

$$\left\| \frac{\mathbf{z}_A}{B} \right\|_1 \leq \frac{1}{2mt}. \quad (32)$$

Every selected vector has coordinate sum zero, so $\mathbf{z}_A$ and the update $\mathbf{a}_A = \mathbf{z}_A/B$ also have coordinate sum zero. This proves the first line of (31).

The weak inequality extends to $\mathcal{K}_t(A)$ by conic linearity. If $\mathbf{y} = \sum_{\mathcal{C}} \beta_{\mathcal{C}} \tilde{\mathbf{g}}_{\mathcal{C}}$ lies in $\mathcal{K}_t(A) \setminus \mathcal{L}_t(A)$, some generator with $\beta_{\mathcal{C}} > 0$ must lie outside $\mathcal{L}_t(A)$; otherwise $\mathbf{y}$ would belong to that subspace. That term makes $\langle \mathbf{a}_A, \mathbf{y} \rangle < 0$. Thus $\mathbf{a}_A$ satisfies Lemma 4.3.

The separation oracle described above takes polynomial time: it sorts the individual utility changes and selects at most $t - |A|$ positive ones. Rational linear programming with a polynomial-time separation oracle is polynomial-time solvable [GLS88]. The successive coordinate minimizations used for tie-breaking require only m additional LP optimizations. The outer basis-building loop appends at most m vectors and tests at most $2m$ signed probes per pass, so it makes $O(m^2)$ calls to the LP subroutine. The initial utility-scaling factor has bit length at most the sum of the bit lengths of the input denominators, plus the bit length of m . The Cramer bound above then gives

polynomial bit length for all selected vertices, B , and the update vectors. Gaussian elimination on these vertices also produces probes of polynomial bit length. In particular,

$$\gamma = \frac{1}{B} \geq 2^{-\text{poly}(\text{input length})}. \quad \square$$

The protocol starts with $\mathbf{p} = \boldsymbol{\mu}$, whose coordinates are $1/m$. Each failed opening enlarges the abort set and applies one update. Commitment failures leave $\mathbf{p}$ unchanged. Each update changes any coordinate by at most $1/(2mt)$, and there are at most t updates. Therefore every reachable coordinate is at least

$$\frac{1}{m} - t \cdot \frac{1}{2mt} = \frac{1}{2m} > 0.$$

Thus every updated center distribution remains a valid probability distribution.

A.3 Choosing the Walk Bias

The punishment bounds now determine a common bias satisfying (9). The following algorithm computes the scaling constants above and this bias from the original instance.

Algorithm 4 Walk bias α and punishment scale

Input: a rational instance (n, m, t, U) with $1 \leq t < n$.

Output: integral vectors $\tilde{\mathbf{g}}_i = d\mathbf{g}_i$ and constants B, γ, α .

- 1: Compute $d = m \prod_{i,j} \text{den}(\text{util}_i(j))$, where den denotes the positive denominator in lowest terms.
 - 2: Set $\tilde{\mathbf{g}}_i \leftarrow d\mathbf{g}_i$ for each $i \in [n]$.
 - 3: Set $H \leftarrow n + \sum_{i,j} |\tilde{g}_i(j)|$ and $B \leftarrow 2m^3 t m! H^m$.
 - 4: Set $\gamma \leftarrow 1/B$ and $\alpha \leftarrow (B^2 + 1)/(2B^2 + 1)$.
 - 5: **return** the integral centered vectors and B, γ, α .
-

Lemma A.5 (A common walk bias). *Algorithm 4 computes in polynomial time a rational $\alpha \in (1/2, 1)$ of polynomial description length. Together with the updates from Algorithm 3, this choice satisfies (9) for every nonempty $A \subseteq \mathcal{C}$ with $|\mathcal{C}| \leq t$ and $\mathbf{g}_{\mathcal{C}} \notin \mathcal{L}_t(A)$.*

Proof. From the setup value of α in Algorithm 4, a direct calculation gives

$$\frac{2\alpha - 1}{1 - \alpha} = \frac{1}{B^2}. \quad (33)$$

This choice also satisfies $1/2 < \alpha < 1$. For every coalition $\mathcal{C}$ of size at most t ,

$$\|\tilde{\mathbf{g}}_{\mathcal{C}}\|_1 \leq \sum_{i \in \mathcal{C}} \|\tilde{\mathbf{g}}_i\|_1 \leq \sum_{i,j} |\tilde{g}_i(j)| \leq H \leq B.$$

For a punishable coalition, Lemma A.4 gives $-\langle \mathbf{a}_A, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle \geq 1/B$. Hence

$$\frac{2\alpha - 1}{1 - \alpha} \|\tilde{\mathbf{g}}_{\mathcal{C}}\|_1 \leq \frac{1}{B^2} B = \frac{1}{B} \leq -\langle \mathbf{a}_A, \tilde{\mathbf{g}}_{\mathcal{C}} \rangle.$$

Dividing both sides by $d > 0$ gives

$$\frac{2\alpha - 1}{1 - \alpha} \|\mathbf{g}_{\mathcal{C}}\|_1 \leq -\langle \mathbf{a}_A, \mathbf{g}_{\mathcal{C}} \rangle,$$

which is (9) for the original utilities.

The scaling constants have polynomial bit length by Lemma A.4. The formula for α therefore also has polynomial bit length and can be evaluated in polynomial time. $\square$

A.4 Efficient Public Parameters

Proposition A.6 (Efficient public protocol parameters). *Given a rational instance (n, m, t, U) with $1 \leq t < n$, a common rational walk bias $\alpha \in (1/2, 1)$ can be computed in polynomial time. For any given nonempty abort set A with $|A| \leq t$, a punishment direction $\mathbf{a}_A$ satisfying Lemma 4.3 can likewise be computed in polynomial time. These choices satisfy (9) for all such A and their compatible coalitions, and have polynomial-size descriptions. One can also decide in polynomial time whether U satisfies Definition 4.5. When it does, the algorithm returns a single order whose restriction to $S = [n] \setminus A$ works for every abort set A with $|A| \leq t$. Otherwise, it reports that the condition fails.*

Proof. Lemma A.2 gives the opening-order decision and construction. Under the rank condition, Algorithm 1 implements the order from Theorem 5.1 in polynomial time. Lemma A.4 supplies the punishment directions and their norm and description-length bounds, and Lemma A.5 supplies the common bias satisfying (9). $\square$

Each update is computed from U , t , and A alone; it does not depend on the current distribution or on the order of earlier aborts. Thus the algorithm specifies one fixed vector for each abort set without storing a table of all abort sets. As in the protocol, set $\mathbf{a}_A = \mathbf{0}$ when $A = \emptyset$ or $|A| > t$.

B Rational Non-uniform Targets

Our protocol extends to rational non-uniform targets with the same resilience. One approach is to initialize the center distribution $\mathbf{p}$ at the desired target and adjust the punishment updates. Here we instead reduce to a uniform target by replacing each outcome with several copies.

Definition B.1 (Expanded utility matrix). Fix a rational target $\mathbf{p}^*$ with $p_j^* > 0$ for each $j \in [m]$, and write $p_j^* = r_j/M$ for positive integers r_j with $M = \sum_j r_j$. The *expanded utility matrix* $\hat{U}$ repeats column j of U exactly r_j times. Each copy (j, k) , $1 \leq k \leq r_j$, has utilities $\hat{U}_{i,(j,k)} = U_{i,j}$.

Lemma B.2 (Preservation of linear relations). *Let $\mathbf{g}_i$ and $\hat{\mathbf{g}}_i$ be the centered utility vectors for U and $\hat{U}$, respectively, each centered at its uniform target. For any real coefficients $\beta_1, \dots, \beta_n$,*

$$\sum_{i=1}^n \beta_i \mathbf{g}_i = \mathbf{0} \iff \sum_{i=1}^n \beta_i \hat{\mathbf{g}}_i = \mathbf{0}. \quad (34)$$

Proof. Averaging over the M copies gives

$$\langle \hat{\boldsymbol{\mu}}, \hat{\mathbf{u}}_i \rangle = \sum_{j=1}^m \frac{r_j}{M} \text{util}_i(j) = \langle \mathbf{p}^*, \mathbf{u}_i \rangle.$$

Therefore each coordinate of the expanded centered vector is

$$\begin{aligned} \hat{g}_i(j, k) &= \text{util}_i(j) - \langle \mathbf{p}^*, \mathbf{u}_i \rangle \\ &= g_i(j) - \langle \mathbf{p}^*, \mathbf{g}_i \rangle. \end{aligned} \quad (35)$$

Fix the coefficients β_i and put $\mathbf{v} := \sum_{i=1}^n \beta_i \mathbf{g}_i$. The corresponding linear combination of expanded vectors has coordinate

$$\sum_{i=1}^n \beta_i \hat{g}_i(j, k) = v_j - \langle \mathbf{p}^*, \mathbf{v} \rangle.$$

If $\mathbf{v} = \mathbf{0}$, every such coordinate is zero. Conversely, if all these coordinates are zero, then $v_j = \langle \mathbf{p}^*, \mathbf{v} \rangle$ for every j , because every outcome has at least one copy. Thus $\mathbf{v}$ is constant across outcomes. Since $\mathbf{v} \in \mathcal{V}_m$, its coordinates sum to zero, so $\mathbf{v} = \mathbf{0}$. This proves (34). Every centered coalition utility vector is a sum of centered party utility vectors. Moving both sides of a coalition-vector equality to one side therefore gives a relation of this form, without changing any restrictions on its coefficients. $\square$

Theorem B.3 (Reduction to a uniform target). *The matrices U and $\hat{U}$ satisfy the abort-safe opening-order condition at threshold t for exactly the same opening orders. If this condition holds, running the uniform protocol for $\hat{U}$ and mapping its output (j, k) to j gives a protocol for target $\mathbf{p}^*$ with negligible correctness error and CSP-fairness at the same threshold, under the same cryptographic assumptions.*

Proof. Since coalition utilities are sums of party utilities, Lemma B.2 also preserves relations among centered coalition utilities. In particular, for $D \subseteq \mathcal{C}$ with $1 \leq |\mathcal{C}| \leq t$, it preserves whether $\mathbf{g}_D = \mathbf{0}$ and whether $-\mathbf{g}_D$ is a nonnegative combination of the generators of $\mathcal{K}_t(D)$. Thus it preserves the test $\mathbf{g}_D \in \mathcal{L}_t(D) \setminus \{\mathbf{0}\}$, so exactly the same opening orders satisfy Definition 4.5.

For correctness, a uniform choice among the M copies maps to outcome j with probability $r_j/M = p_j^*$. Merging the probabilities of copies cannot increase the ℓ_1 error, so correctness follows from Theorem 4.2.

For CSP-fairness, the interaction is unchanged, and each copy gives every party the same utility as its original outcome. Every coalition's payoff is therefore unchanged under both honest play and every deviation. CSP-fairness follows from Theorem 4.7, with the same error and corruption threshold. $\square$

In particular, the lemma preserves centered rank, so $\text{rank}(G) \geq n - 1$ still gives maximal resilience by Theorem 5.1. For targets with zero probabilities, restrict U to the target's support and check the opening-order condition there.

C More Examples and Complete Characterization for $n = 3$

C.1 Rank Condition is Not Necessary for Abort-Safe Opening-Order Condition

The abort-safe opening-order condition can hold at the maximal threshold even when the rank condition $\text{rank}(G) \geq n - 1$ fails. Consider $n = m = 4$ and $t = 3$, with utility matrix

$$U = \begin{pmatrix} 19 & 17 & 11 & 1 \\ 13 & 15 & 13 & 7 \\ 7 & 13 & 15 & 13 \\ 1 & 11 & 17 & 19 \end{pmatrix}. \quad (36)$$

Every row has average 12, so the matrix of centered utility rows is

$$G^\top = U \left(I_4 - \frac{1}{4} \mathbf{1}\mathbf{1}^\top \right) = \begin{pmatrix} 7 & 5 & -1 & -11 \\ 1 & 3 & 1 & -5 \\ -5 & 1 & 3 & 1 \\ -11 & -1 & 5 & 7 \end{pmatrix}, \quad (37)$$

where I_4 is the 4×4 identity matrix. Put

$$\mathbf{a} = (-2, 2, 2, -2)^\top, \quad \mathbf{b} = (-3, -1, 1, 3)^\top, \quad (t_1, t_2, t_3, t_4) = (-3, -1, 1, 3).$$

The centered utility vector of party i satisfies

$$\mathbf{g}_i = \mathbf{a} + t_i \mathbf{b}. \quad (38)$$

Since $\mathbf{a}$ and $\mathbf{b}$ are linearly independent and the t_i are distinct, their span is $\text{span}\{\mathbf{a}, \mathbf{b}\}$. Hence $\text{rank}(G) = 2 < 3$, so the rank condition does not apply.

Nevertheless, every opening order is abort-safe at threshold 3. Indeed,

$$\langle \mathbf{a}, \mathbf{b} \rangle = 0, \quad \langle \mathbf{a}, \mathbf{a} \rangle = 16, \quad (39)$$

and every nonempty coalition $\mathcal{C}$ satisfies

$$\langle \mathbf{a}, \mathbf{g}_{\mathcal{C}} \rangle = \sum_{i \in \mathcal{C}} \langle \mathbf{a}, \mathbf{g}_i \rangle = 16|\mathcal{C}| > 0. \quad (40)$$

Thus moving in direction $-\mathbf{a}$ strictly decreases every nonempty coalition's utility. In particular, for any nonempty $D \subsetneq [4]$, every nonzero vector in $\mathcal{K}_3(D)$ has positive inner product with $\mathbf{a}$. The cone cannot contain both a nonzero vector and its negative, so

$$\mathcal{L}_3(D) = \mathcal{K}_3(D) \cap (-\mathcal{K}_3(D)) = \{\mathbf{0}\}. \quad (41)$$

For every abort set X and remaining coalition member i in Definition 4.5, the set $X \cup \{i\}$ is nonempty. Therefore its unpunishable subspace is trivial, and the premise of the opening-order condition never holds. Every opening order satisfies the condition, including after earlier aborts.

Consequently, Theorems 4.2 and 4.7 give a correct, CSP-fair protocol at the maximal threshold, under the same commitment assumption as in the construction. This example shows that the general opening-order condition is strictly more permissive than the rank condition.

C.2 An Obstruction with $k = 3$

The impossibility criterion can require three score vectors even when every admissible pair has zero product average. Consider $n = m = 6$ and $t = 5$, with utility matrix

$$U = \begin{pmatrix} 5 & 4 & 2 & 1 & 2 & 4 \\ 3 & 6 & 2 & 3 & 4 & 0 \\ 3 & 4 & 6 & 3 & 0 & 2 \\ 1 & 2 & 4 & 5 & 4 & 2 \\ 3 & 2 & 0 & 3 & 6 & 4 \\ 3 & 0 & 4 & 3 & 2 & 6 \end{pmatrix}. \quad (42)$$

Every row has average 3, and the centered utility vectors are $\mathbf{a}, \mathbf{c}, \mathbf{b}, -\mathbf{a}, -\mathbf{b}, -\mathbf{c}$, where

$$\begin{aligned} \mathbf{a} &= (2, 1, -1, -2, -1, 1)^\top, \\ \mathbf{b} &= (0, 1, 3, 0, -3, -1)^\top, \\ \mathbf{c} &= (0, 3, -1, 0, 1, -3)^\top. \end{aligned} \quad (43)$$

These three nonzero vectors are pairwise orthogonal. In the partition of Equation (16), take $Y = \emptyset$ and

$$P_1 = \{1, 4\}, \quad P_2 = \{3, 5\}, \quad P_3 = \{2, 6\}.$$

For each $\ell \in [3]$, the sets $A_\ell = C_\ell = [6] \setminus P_\ell$ consist of the other two complete pairs and have size $4 \leq t$. The coalitions generating $\mathcal{K}_5(A_1)$ are A_1 itself, whose centered utility vector is zero, and

the two sets obtained by adjoining party 1 or party 4, whose centered utility vectors are $\mathbf{a}$ and $-\mathbf{a}$. The same calculation for the other two pairs gives

$$\begin{aligned}\mathcal{K}_5(A_1) &= \mathcal{L}_5(A_1) = \text{span}\{\mathbf{a}\}, \\ \mathcal{K}_5(A_2) &= \mathcal{L}_5(A_2) = \text{span}\{\mathbf{b}\}, \\ \mathcal{K}_5(A_3) &= \mathcal{L}_5(A_3) = \text{span}\{\mathbf{c}\}.\end{aligned}$$

Thus $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = (\mathbf{a}, \mathbf{b}, \mathbf{c})$ satisfies (17).

To see why two score vectors cannot suffice, first observe that $\mathbf{g}_{[6]} = \mathbf{0}$. For every $S \subseteq [6]$, this gives

$$\mathcal{K}_5(S) = \text{cone}(\{-\mathbf{g}_j : j \notin S\}).$$

Indeed, every admissible coalition vector is the negative sum of its omitted parties' centered utilities, and each $-\mathbf{g}_j$ on the right is supplied by the coalition $[6] \setminus \{j\}$. The six centered utility vectors form three opposite pairs on independent, orthogonal axes. Consequently, $\mathcal{L}_5(S)$ is spanned by exactly those axes whose complete pair lies outside S .

Now consider any partition $[6] = Y \cup P_1 \cup P_2$ into disjoint sets, with P_1, P_2 nonempty, and any admissible pair of score vectors. We have

$$\mathbf{v}_1 \in \mathcal{K}_5([6] \setminus P_1) \cap \mathcal{L}_5(P_2), \quad \mathbf{v}_2 \in \mathcal{K}_5([6] \setminus P_2) \cap \mathcal{L}_5(P_1).$$

Every nonzero component of $\mathbf{v}_1$ must come from a pair with a party in P_1 and no party in P_2 . Symmetrically, every nonzero component of $\mathbf{v}_2$ comes from a pair with a party in P_2 and no party in P_1 . The two vectors therefore use different axes. By orthogonality,

$$\frac{1}{6} \sum_{j=1}^6 v_1(j)v_2(j) = 0.$$

For the three-block partition above, however, the scores $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = (\mathbf{a}, \mathbf{b}, \mathbf{c})$ give

$$\frac{1}{6} \sum_{j=1}^6 v_1(j)v_2(j)v_3(j) = 2. \tag{44}$$

Theorem 6.1 therefore rules out every correct polynomial-round protocol that is CSP-fair against PPT rushing semi-malicious coalitions of size at most five. Thus three directions witness the obstruction even though every admissible pair has zero product average.

C.3 Complete Characterization for Three Parties

For three parties, our characterization is tight against size-two corruption.

Proposition C.1 (Three-party characterization). *Fix $n = 3$, $t = 2$, and the utility matrix U . If U does not satisfy the abort-safe opening-order condition, then the impossibility criterion of Theorem 6.1 applies with $k = 2$. Consequently, assuming one-way functions exist, a correct polynomial-round protocol that is CSP-fair against PPT rushing semi-malicious coalitions of size at most two exists if and only if U satisfies the abort-safe opening-order condition.*

Proof. If $\text{rank}(G) \geq 2$, then Theorem 5.1 gives an abort-safe opening order. If $\text{rank}(G) = 0$, every centered coalition utility vector is zero, so every opening order works.

Suppose that $\text{rank}(G) = 1$, and choose a nonzero rational vector $\mathbf{g} \in \mathcal{V}_m$ spanning the centered utility vectors. If the three vectors $\mathbf{g}_i$ are all nonnegative multiples of $\mathbf{g}$, or all nonpositive multiples, every coalition-utility cone lies in a single ray. Its unpunishable subspace is therefore trivial, and again every opening order works.

Otherwise, after relabeling parties and choosing the sign of $\mathbf{g}$, write

$$\mathbf{g}_1 = a\mathbf{g}, \quad \mathbf{g}_2 = b\mathbf{g}, \quad \mathbf{g}_3 = -c\mathbf{g}, \quad 0 \leq a \leq b, \quad b, c > 0, \quad (45)$$

where a, b, c are rational coefficients. The case $a = 0$ allows one party to be indifferent among all outcomes. The singleton cones are

$$\begin{aligned} \mathcal{K}_2(\{1\}) &= \text{cone}(\{a\mathbf{g}, (a+b)\mathbf{g}, (a-c)\mathbf{g}\}), \\ \mathcal{K}_2(\{2\}) &= \text{cone}(\{b\mathbf{g}, (a+b)\mathbf{g}, (b-c)\mathbf{g}\}), \\ \mathcal{K}_2(\{3\}) &= \text{cone}(\{-c\mathbf{g}, (a-c)\mathbf{g}, (b-c)\mathbf{g}\}). \end{aligned}$$

Each has nontrivial lineality exactly when its generators include both positive and negative multiples of $\mathbf{g}$. In that case its lineality space is $\text{span}\{\mathbf{g}\}$. For a two-party set D , the cone $\mathcal{K}_2(D)$ has only the generator $\mathbf{g}_D$, so $\mathcal{L}_2(D) = \{\mathbf{0}\}$. It therefore suffices to check the opening-order condition when $X = \emptyset$. After an earlier abort, the premise of (12) cannot hold.

Case 1: $c \leq a$. All generators of $\mathcal{K}_2(\{1\})$ and $\mathcal{K}_2(\{2\})$ are nonnegative multiples of $\mathbf{g}$, so

$$\mathcal{L}_2(\{1\}) = \mathcal{L}_2(\{2\}) = \{\mathbf{0}\}.$$

Use the order $\pi = (3, 1, 2)$. Party 3 opens before every party outside any proper coalition containing it, and parties 1 and 2 have trivial unpunishable subspaces. Hence the order is abort-safe.

Case 2: $a < c = b$. Use $\pi = (1, 3, 2)$. Party 1 opens first, so any proper coalition containing it has an honest party scheduled later. For party 3, the only admissible coalition that contains it and every later party is $\{2, 3\}$. This coalition has centered utility $\mathbf{g}_{\{2,3\}} = (b-c)\mathbf{g} = \mathbf{0}$. Finally, $\mathcal{L}_2(\{2\}) = \{\mathbf{0}\}$, so party 2 can safely open last. Thus this order is also abort-safe.

Case 3: $a < c < b$. The generators $(a+b)\mathbf{g}$ and $(a-c)\mathbf{g}$ have opposite signs, as do $-c\mathbf{g}$ and $(b-c)\mathbf{g}$. Hence

$$\mathcal{L}_2(\{1\}) = \mathcal{L}_2(\{3\}) = \text{span}\{\mathbf{g}\}.$$

Take the partition

$$P_1 = \{1\}, \quad P_2 = \{3\}, \quad Y = \{2\}.$$

Then $A_1 = \{3\}$, $C_1 = \{2, 3\}$, $A_2 = \{1\}$, and $C_2 = \{1, 2\}$. Choose

$$\mathbf{v}_1 = \mathbf{g}_{C_1} = (b-c)\mathbf{g}, \quad \mathbf{v}_2 = \mathbf{g}_{C_2} = (a+b)\mathbf{g}.$$

Each score belongs to $\mathcal{K}_2(C_\ell)$ because it is that coalition's centered utility vector, and to $\mathcal{L}_2(A_\ell)$ by the displayed lineality calculation. Thus (17) holds, and

$$\eta = \frac{(b-c)(a+b)}{m} \langle \mathbf{g}, \mathbf{g} \rangle > 0.$$

Theorem 6.1 applies with $k = 2$.

Case 4: $c > b$. Now $(a+b)\mathbf{g}$ is a positive generator in both $\mathcal{K}_2(\{1\})$ and $\mathcal{K}_2(\{2\})$, while $(a-c)\mathbf{g}$ and $(b-c)\mathbf{g}$, respectively, are negative generators. Therefore

$$\mathcal{L}_2(\{1\}) = \mathcal{L}_2(\{2\}) = \text{span}\{\mathbf{g}\}.$$

Take

$$P_1 = \{1\}, \quad P_2 = \{2\}, \quad Y = \{3\}.$$

Then $A_1 = \{2\}$, $C_1 = \{2, 3\}$, $A_2 = \{1\}$, and $C_2 = \{1, 3\}$. The scores

$$\mathbf{v}_1 = \mathbf{g}_{C_1} = (b - c)\mathbf{g}, \quad \mathbf{v}_2 = \mathbf{g}_{C_2} = (a - c)\mathbf{g}$$

satisfy (17), and

$$\eta = \frac{(c - b)(c - a)}{m} \langle \mathbf{g}, \mathbf{g} \rangle > 0.$$

Again, Theorem 6.1 applies with $k = 2$.

In each of the last two cases, the singleton blocks also show directly why no opening order works. Let i, j be their two parties, with i scheduled before j , and take $\mathcal{C} = [3] \setminus \{i\}$. The calculations above give $\mathbf{g}_{\mathcal{C}} \neq \mathbf{0}$ and $\mathcal{L}_2(\{j\}) = \text{span}\{\mathbf{g}\}$. Thus $\mathbf{g}_{\mathcal{C}} \in \mathcal{L}_2(\{j\}) \setminus \{\mathbf{0}\}$, but the only party outside $\mathcal{C}$ is i , which opens before j . This violates (12) at $X = \emptyset$.

The four cases are exhaustive. In particular, for the mixed-sign profile (45) with $\text{rank}(G) = 1$, the opening-order condition holds exactly when

$$c \leq a \quad \text{or} \quad c = b. \tag{46}$$

Whenever the condition holds, Theorems 4.2 and 4.7 give a correct polynomial-round CSP-fair protocol under the stated commitment assumption. Whenever it fails, one of the two-score witnesses above rules out every such protocol. This proves the characterization. $\square$

AI Use Disclosure

The authors used GPT-5.6 `xhigh` to generate an initial draft of the TikZ code for Figure 1, check grammar, and improve language clarity. The authors substantially revised the generated figure code, adding labels, adjusting the layout to remove overlaps, and improving its visual presentation. Discussions with the same model also provided some insights that helped the authors develop the online punishment update in Algorithm 3 in Appendix A. The prompt used to generate the initial figure code was:

create this diagram in tikz. a left center 0 node connected to three horizontal paths, with transitions labeled p_i, alpha, and 1-alpha, ellipses for skipped states, double-circle endpoints, and labels output 1-3