Game Theoretic Notions of Fairness in Multi-Party Coin Toss*

Kai-Min Chung[2], Yue Guo[1], Wei-Kai Lin[1], Rafael Pass[1,3], and Elaine Shi[1]

[1] Cornell University, Ithaca, NY, USA
{yueguo,wklin,rafael,elaine}@cs.cornell.edu
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan
kmchung@iis.sinica.edu.tw
[3] CornellTech, New York, NY, USA

Abstract. Coin toss has been extensively studied in the cryptography literature, and the well-accepted notion of fairness (henceforth called *strong fairness*) requires that a corrupt coalition cannot cause non-negligible bias. It is well-understood that two-party coin toss is impossible if one of the parties can prematurely abort; further, this impossibility generalizes to multiple parties with a corrupt majority (even if the adversary is computationally bounded and fail-stop only).

Interestingly, the original proposal of (two-party) coin toss protocols by Blum in fact considered a weaker notion of fairness: imagine that the (randomized) transcript of the coin toss protocol defines a winner among the two parties. Now Blum's notion requires that a corrupt party cannot bias the outcome in its favor (but self-sacrificing bias is allowed). Blum showed that this weak notion is indeed attainable for two parties assuming the existence of one-way functions.

In this paper, we ask a very natural question which, surprisingly, has been overlooked by the cryptography literature: can we achieve Blum's weak fairness notion in *multi-party* coin toss? What is particularly interesting is whether this relaxation allows us to circumvent the corrupt majority impossibility that pertains to strong fairness. Even more surprisingly, in answering this question, we realize that it is not even understood how to define weak fairness for multi-party coin toss. We propose several natural notions drawing inspirations from game theory, all of which equate to Blum's notion for the special case of two parties. We show, however, that for multiple parties, these notions vary in strength and lead to different feasibility and infeasibility results.

1 Introduction

The study of coin toss protocols was initiated in Blum's ground-breaking work [16]. Consider the following scenario: Alice and Bob had concurrent and independent

 This work is supported in part by NSF grants CNS-1514261 and CNS-1561209. This is the conference version. The full version of this paper is available online.

results that solved a difficult open question in cryptography. Both submitted their papers to the prestigious Theory of Cryptography Conference (TCC) 2018 conference with the most amazing program committee (PC). The wise PC urged Alice and Bob to merge their results into one paper and provided them with a single presentation slot at the conference. Now Alice and Bob would like to toss a random coin to decide who goes to the most fabulous conference venue ever, Goa, and present the paper. Since Alice and Bob are not in the same room, they would like to complete the coin toss by sending messages to each other (slowly) over the Ethereum blockchain, such that anyone who observes the transcript can determine the outcome of the coin flip. Now either party would like to make sure that he/she has a fair chance of winning even when the other cheats and deviates from the protocol. The academic literature has since referred to Blum's notion of fairness as *weak fairness*; and Blum showed that assuming the existence of one-way functions a weakly-fair, 2-party coin toss protocol can be constructed [16]. Interestingly, however, the vast majority of subsequent cryptography literature has focused on a stronger notion of fairness than Blum's, that is, a corrupt party cannot bias the outcome of the coin toss — henceforth we refer to this notion as *strong fairness* [19]. It is not difficult to see that a strongly fair coin toss protocol must also be weakly fair; but not the other way round. In particular, a weakly fair protocol allows a corrupt party to bias the outcome of the remaining honest party — but the bias must not be in the corrupt party's favor. Unfortunately for the strongly fair notion, Cleve's celebrated result [19] proved its impossibility in a 2-party setting even for computationally bounded, fail-stop adversaries.

In this paper, we consider multi-party extensions of Blum's notion of weak fairness. We ask a very natural question that seems to have been overlooked by the literature so far:

> *Can we achieve Blum's weak fairness notion in multi-party coin toss protocols?*

By contrast, the strong fairness notion has been extensively studied in the multi-party context [12,27]. Well-known results tell us that the strong notion is attainable assuming honest majority and existence of one-way functions. On the other hand, Cleve's 2-party impossibility extends to multiple parties with a corrupt majority [19]. Therefore, a more refined question is

> *Can we overcome Cleve's impossibility for corrupt majority multi-party coin toss with weak fairness?*

Of course, to answer the above questions, we must first answer

> *How do we even define weak fairness in multi-party coin toss protocols?*

Intriguingly, even the definition itself is non-trivial! In this paper, we propose several natural notions of fairness that are inspired by the line of work on game theory [15, 35, 36, 48]. Interestingly, all of these notions equate to Blum's notion for the special case of 2 parties; however, in general, they differ in strengths for multiple parties and thus lead to differing feasibility and infeasibility results.

1.1 Our Results and Contributions

Consider the following scenario: n parties would like to play a 1-bit roulette game over the Internet: First, each party puts down 1 Ether as stake and places a *publicly visible*[4] bet (also referred to as the party's *preference*) on one of the bits $b \in \{0, 1\}$. Without loss of generality *we assume that not everyone bets on the same bit*. Next, they run an n-party coin toss protocol by exchanging messages over the Ethereum blockchain, and transcript of the protocol determines an outcome bit. Now, those who betted correctly are called winners; and those who betted wrongly are called losers. Finally, every loser loses its stake to the house (e.g., owner of the smart contract); and each winner gets paid 1 Ether by the house. We require that *in an honest execution, each bit is chosen with probability* $1/2$. Henceforth in the paper for simplicity we shall think of the Ethereum blockchain as a broadcast medium with identifiable abort, i.e., a public bulletin board that allows parties to post messages.

How should we define fairness for this 1-bit roulette game? Cryptography and game theory provide different answers. The standard notion from cryptography is again strong fairness [19], that is, any corrupt coalition should not be able to bias the outcome by more than a negligible amount. As mentioned strong fairness is unattainable under a corrupt majority even for fail-stop adversaries [19]. Most of game theory, on the other hand, considers (computational) Nash Equilibrium [48], that is, no corrupt *individual* can noticeably improve its expected reward by deviating, assuming that everyone else is playing honestly. Although Nash Equilibrium is indeed attainable by adopting a standard, strongly fair multi-party coin toss protocol that tolerates deviation by any single party [27], such a notion might be too weak. In particular, no guarantee is provided when two or more parties collude (e.g., in cryptocurrency applications, an individual user can always make up any number of pseudonyms and control the majority in a game). Therefore we would like to explore notions in between that allow us to resist majority coalitions and provide meaningful fairness guarantees in practical applications. In this paper, we define several notions of fairness — all of them equate to Blum's notion [16] for the special case of 2 parties. Thus for all of our notions, in the 2-party case Blum's result applies: assuming one-way functions, all notions are attainable against malicious, computationally bounded adversaries that control one of the two parties; moreover, for fail-stop adversaries, all our notions are attainable against even unbounded adversaries that control one of the two parties.

Henceforth for our fairness notions, we are concerned about feasibility and infeasibility results for 3 or more parties, and particularly for the case of *corrupt majority* (since for honest majority, feasibility is known even for strong fairness, against malicious, computationally bounded adversaries, due to the celebrated result by Goldreich et al. [27]). As a final remark before we introduce our no-

[4] Unless otherwise noted, we consider public preference profiles. For completeness, however, we present results for private preference profiles in the appendices, Section 7.

tions, all our notions (as well as Blum's notion) can be easily ruled out for computationally unbounded, malicious adversaries [33, 44].

1.2 Maximin Fairness

Definition. A natural notion, which seems to be a good fit for cryptocurrency applications, is to require the following: an honest Alice should not be harmed even when everyone else is colluding against her. In other words, any individual's expected reward should not noticeably decrease (relative to an all-honest execution) even when all others are colluding against her. This notion has a game theoretic interpretation: the honest strategy maximizes a player's worst-case expected payoff (even when everyone else is colluding against her); and moreover, by playing honest, the player's worst-case expected payoff is not noticeably worse than an all-honest execution. For maximin fairness, we present a complete characterization of feasibilities/infeasibilties.

Feasibility and infeasibility for almost unanimous preference profiles. For 3 or more parties, if everyone agrees in preference except one party, we say that the preference profile is *almost unanimous*. When the preference profile is almost unanimous, maximin fairness is possible for fail-stop adversaries and without relying on cryptographic assumptions. Recall that a fail-stop adversary may prematurely abort from the protocol but would otherwise follow the honest protocol [19]. The corresponding protocol is very simple: without loss of generality assume that one party prefers 0 (called the 0-supporter) and all others prefer 1 (called the 1-supporters). Now, the 0-supporter chooses a random bit and broadcasts it. If the broadcast indeed happens, the bit broadcast is declared as the outcome. Otherwise, the outcome is defined to be 1.

We then prove that for an almost unanimous preference profile, maximin fairness is impossible for malicious adversaries even when allowing cryptographic assumptions. This result is somewhat counter-intuitive in light of the earlier feasibility for fail-stop (and the proof rather non-trivial too). In particular, in most of the cryptography literature, we are familiar with techniques that compile fail-stop (or semi-honest) protocols to attain full, malicious security [12, 27] — but these compilation techniques *do not preserve maximin fairness* and thus are inapplicable here.

Note that for the special case of 3 parties, unless everyone has the same preference any preference profile is almost unanimous — thus for the case of 3 parties we already have a complete characterization. For 4 or more parties, we need to consider the case when the preference profile is more mixed.

Infeasibility for amply divided preference profiles. If there are at least two 0-supporters and at least two 1-supporters, we say that the parties have an *amply divided* preference profile. Note that for 3 or more parties, unless everyone has the same preference, then every preference profile is either almost unanimous or amply divided. For an amply divided preference profile, we show infeasibility even against computationally bounded, fail-stop adversaries by reduction to Cleve's impossibility result for strong fairness [19].

We summarize our results for maximin fairness in the following theorems — although not explicitly noted, all theorems are concerned about an adversary that may control up to $n-1$ players.

Theorem 1 (Maximin fairness: upper bound (informal)). *For any $n \geq 3$ and any almost unanimous preference profile, there is an n-party coin toss protocol that achieves maximin fairness against fail-stop and computationally unbounded adversaries.*

Theorem 2 (Maximin fairness: lower bound (informal)). *For any $n \geq 3$ and any almost unanimous preference profile, no n-party coin toss protocol can achieve maximin fairness against malicious and even polynomially bounded adversaries. Further, for any $n \geq 4$ and any amply divided preference profile, no n-party coin toss protocol can achieve maximin fairness against fail-stop and even polynomially bounded adversaries.*

Summary. While maximin fairness appears to provide strong guarantees in cryptocurrency and smart contract applications, we showed rather broad infeasibility results. Nonetheless it gives us a glimpse of hope: for the case of almost unanimous preference profiles and fail-stop adversaries, we are able to achieve positive results for corrupt majority while strong fairness cannot! We thus continue to explore alternative notions in hope of finding one that leads to broader feasibility results. Our high-level idea is the following: earlier, maximin fairness aims to rule out coalitions that *harm honest parties*; instead we now consider notions that rule out coalitions capable of *improving its own wealth* — this gives rise to two new notions, cooperative-strategy-proof fairness and Strong Nash Equilibrium, as we discuss subsequently in Sections 1.3 and 1.4.

1.3 Cooperative-Strategy-Proof Fairness

Definition. Cooperative-strategy-proof (CSP) fairness requires that no deviation by a corrupt coalition of size up to $n-1$ can noticeably improve the coalition's total expected reward relative to an honest execution. It is not difficult to see that CSP fairness is equivalent to maximin fairness for zero-sum cases: when exactly half prefer 0 and half prefer 1. However, the two notions are incomparable in general.

Feasibility for almost unanimous preference profiles. When almost everyone prefers the same bit except for one party, we show that the following simple protocol achieves CSP fairness against malicious adversaries. For simplicity, our description below assumes an ideal commitment functionality $\mathcal{F}_{\text{idealcomm}}$ — but this idealized oracle can be replaced with suitable non-malleable concurrent commitment schemes [42, 43] with some additional work. Without loss of generality we assume that a single party prefers 0 and everyone else prefers 1: First, everyone picks a random bit upfront and commits the bit to $\mathcal{F}_{\text{idealcomm}}$. In round 0, the single 0-supporter opens its committed bit and broadcasts it. In round 1,

everyone else opens its committed bit and broadcasts the opening. The outcome is defined to be 0 if one or more 1-supporter(s) aborted; else it is defined to be the XOR of all bits that have been correctly opened.

Finally, for fail-stop adversaries, a variant of the above protocol without commitment can achieve CSP fairness against even unbounded adversaries.

Infeasibility for amply divided preference profiles. For any amply divided preference profile, we prove that it is impossible to achieve CSP fairness against even fail-stop, polynomially bounded adversaries.

We summarize results for CSP fairness in the following theorem.

Theorem 3 (CSP fairness (informal)). *For any almost unanimous preference profile, it is possible to attain CSP fairness against fail-stop, unbounded adversaries, and against malicious, polynomially bounded adversaries assuming one-way permutations. By contrast, for any amply divided preference profile, it is impossible to attain CSP fairness against even fail-stop, polynomially bounded adversaries.*

1.4 Strong Nash Equilibrium

Due to earlier impossibility results for maximin fairness and CSP fairness, we ask if there is a notion for which we can enjoy broad feasibility. To this end we consider a fairness notion inspired by Strong Nash Equilibrium (SNE) [35], henceforth referred to as SNE fairness. SNE fairness requires that no deviation by a coalition can improve every coalition member's expected reward. It is not difficult to see that for SNE fairness, we only need to resist *unanimous* coalitions, i.e., coalitions in which every member prefers the same bit. Further, SNE fairness is also strictly weaker than CSP fairness in general.

We show that a simple dueling protocol achieves SNE fairness against malicious (but polynomially bounded) adversaries: pick two parties with opposing preferences (i.e., pick the two with smallest possible party identifiers), and then have the two run Blum's weak coin toss protocol. Further, the computational assumptions can be removed for fail-stop adversaries and thus SNE fairness can be guaranteed unconditionally for the fail-stop case. We summarize our results on SNE fairness in the following theorem.

Theorem 4 (SNE fairness (informal)). *For any $n \geq 3$ and any preference profile: 1) there is an n-party coin toss protocol that achieves SNE fairness against malicious, polynomially-bounded adversaries assuming the existence of one-way permutations; and 2) there is an n-party coin toss protocol that achieves SNE fairness against fail-stop, unbounded adversaries.*

Alternative formulation: cooperative-coalition-proof fairness. While SNE fairness aims to rule out coalitions that improve every coalition member's wealth, an alternative notion would be to resist *self-enforcing* coalitions that aim to improve the coalition's overall wealth. In particular, a coalition is said to be self-enforcing iff no *self-enforcing* sub-coalition can gain by deviating from the

coalition's original strategy. Such coalitions are stable and will not implode due to internally misaligned incentives. We formalize this notion in our online full version [18] which we call *cooperative-coalition-proof fairness* (CCP fairness). Since CCP fairness considers complex coalition and sub-coalition behavior, we can no longer use the familiar protocol execution model used in the standard cryptography literature — we instead propose a new, suitable protocol execution model that allows us to characterize complex coalition structures. Our CCP fairness notion is inspired by the notion of coalition-proof Nash equilibrium (CPNE) [15] in game theory — but unlike CPNE which considers self-enforcing coalitions that seek to improve every member's gain, our CCP notion considers self-enforcing coalitions that seek to improve its overall gain, and thus our notion is stronger (i.e., demands stronger solution concepts).

Although for general games, SNE fairness and CCP fairness are incomparable, we prove that for the special case of multi-party coin toss, the two notions are in fact equivalent! In this context both notions effectively rule out *unanimous* coalitions where everyone prefers the same outcome.

1.5 Technical Highlight

Conceptual, definitional contributions. First, we make a conceptual contribution by introducing several natural, game-theoretical notions of fairness for multi-party coin toss — our work thus opens a new avenue for connecting game theory and cryptography. Earlier efforts at connecting game theory and multi-party computation typically model the correctness and/or confidentiality of multi-party protocols as a game (see Section 8 for more discussions), whereas we consider a model in which each party independently declares the utility for various outcomes.

A new framework for proving lower bounds. Our upper bounds are simple and intuitive in hindsight (but note that several upper bounds were not immediately obvious to us in the beginning). Our main lower bound results, however, are rather non-trivial to prove. The most non-trivial proofs are 1) the impossibility of maximin fairness for almost unanimous preference profiles, against malicious, computationally bounded adversaries; and 2) the impossibility of CSP fairness for amply divided preference profiles, this time against fail-stop and computationally bounded adversaries.

We develop a new proof framework and apply this framework to rule out both maximin fairness (for almost unanimous, malicious) and CSP fairness (for amply divided, fail-stop)[5]. In this proof framework, we would carefully group nodes into three partitions such that we can view the execution as a 3-party protocol (between the partitions). In both impossibility proofs, we show that the requirements of maximin or CSP fairness imposes a set of conditions that are by nature self-contradictory and thus cannot co-exist.

[5] Interestingly, later in our online full version [18] we again reuse the same proof framework to prove lower bounds for private-preference protocols too.

Since the lower bound proofs are highly non-trivial, to help the reader we give an informal narrative of the maximin proof in Section 4.4. Then, in Section 5.3, we intuitively explain the additional challenges that arise for ruling out CSP fairness (for amply divided, fail-stop) — this proof is even more challenging than maximin fairness (for almost unanimous, malicious) partly because we need to rule out even fail-stop adversaries in this case. The full formal proofs are deferred to the appendices due to lack of space.

2 Preliminaries

2.1 Protocol Execution Model

A protocol is a system of Interactive Turing Machines (ITMs) where each ITM is also referred to as a *party* or a *player*. Each party is either *honest* or *corrupt*. Honest parties correctly follow the protocol to the end without aborting. Corrupt parties, on the other hand, are controlled by an adversary $\mathcal{A}$. Corrupt parties forward all received messages to $\mathcal{A}$ and send messages or abort based on $\mathcal{A}$'s instructions. In this way, we can view the set of all corrupt parties as a single coalition that collude with one another.

A protocol's execution is parametrized by a security parameter $\kappa \in N$ that is public known to all parties including the adversary $\mathcal{A}$. A protocol's execution may be randomized where all parties and the adversary $\mathcal{A}$ receive and consume a string of random bits.

We assume a *round-based* execution model. In each round, every honest party can perform any polynomial in κ amount of computation. At the end of the round, every party may broadcast a message whose length must be polynomial in κ as well. We assume a *synchronous broadcast* medium (with identifiable abort) for parties to communicate with each other. Messages sent by honest parties in round r will be delivered to all honest parties at the beginning of round $r + 1$. If a party i aborts the protocol in round r without sending any message, then all honest parties can detect such abort by detecting the absence of i's message at the beginning of round $r + 1$. As an example, one can imagine that parties communicate by posting messages to a public blockchain such as Bitcoin [26, 47, 50, 51][6].

2.2 Corruption Models

The adversary can corrupt any number of parties. Without loss of generality, we assume that for any fixed adversary algorithm $\mathcal{A}$, the set of parties it wants to corrupt is deterministically encoded in the description of $\mathcal{A}$ (i.e., for any fixed adversary $\mathcal{A}$, there is no randomness in the choice of the corrupt coalition). We

[6] Although a blockchain typically requires honest majority assumptions to retain security, the parties involved in the coin-toss protocol can be majority corrupt.

assume that the adversary is capable of a *rushing* attack[7], i.e., in any round r, the adversary is allowed to view messages sent by honest parties in round r, before deciding what messages corrupt parties will send in round r.

Depending on the adversary's capability, we say that the adversary is fail-stop, semi-malicious, or malicious. More formally, let Π denote the honest protocol under consideration. An adversarial algorithm $\mathcal{A}$ is said to be *fail-stop*, *semi-malicious*, or *malicious* w.r.t. Π iff the following holds:

- *Fail-stop:* Corrupt nodes always follow the honest protocol but may abort in the middle of the protocol. The decision to abort (or not) can depend on the corrupt parties' view in the protocol so far.
- *Malicious:* The adversary can make corrupt parties deviate arbitrarily from the prescribed protocol, including sending arbitrary messages, choosing randomness arbitrarily, and aborting prematurely.

2.3 Additional Notations and Assumptions

Throughout the paper, we assume that the number of parties is polynomially bounded, i.e., $n = \mathsf{poly}(\kappa)$ for some polynomial function $\mathsf{poly}(\cdot)$. We consider protocols that terminate in polynomially many rounds. Specifically, there exists some polynomial $R(\cdot)$ that denotes the round complexity of the protocol, such that with probability 1, honest parties complete execution in $R(\kappa)$ even in the presence of any (possibly computationally unbounded) adversary controlling any corrupt coalition.

We say that a function $\nu(\cdot)$ is a *negligible* function iff for every polynomial function $p(\cdot)$, there exists some $\kappa_0 \in \mathbb{N}$ such that $\nu(\kappa) \leq 1/p(\kappa_0)$ for all $\kappa \geq \kappa_0$.

3 Definitions: Multi-Party Coin Toss

As in the standard cryptography literature, we model protocol execution as a system of Interactive Turing Machines. We consider a synchronous model with a broadcast medium. Messages broadcast by honest parties in the current round are guaranteed to be delivered at the beginning of the next round. We assume *identifiable abort*, that is, failure to send a message is publicly detectable.

We assume that the adversary, denoted $\mathcal{A}$, can control any number of parties. Without loss of generality, we assume that the set of parties $\mathcal{A}$ wants to corrupt is hard-wired in the description of $\mathcal{A}$. We assume a simultaneous messaging model with the possibility of *rushing attacks*, that is, the adversary can observe honest nodes' messages before deciding corrupt nodes' actions (including what messages to send and whether to abort) in any round.

[7] We note that in a simultaneous message model where the adversary is not capable of rushing attacks, even the standard notion of (strong) fairness [19] (which is stronger than all notions considered in this paper) is trivial to achieve for 2-party or multi-party coin toss, even against any majority corrupt coalition.

Recall that a *fail-stop* party is one that could abort prematurely but would otherwise follow the honest protocol. By contrast, a *malicious* party is one that can deviate arbitrarily from the honest protocol.

3.1 Multi-Party Coin Toss

Preference profile. Suppose that each party starts with a *preference* among the two outcomes 0 and 1. The vector of all parties' preferences, denoted $\mathcal{P} := \{0,1\}^n$, is referred to as a preference profile. We sometimes refer to a party that prefers 1 as a 1-*supporter* and we refer to one that prefers 0 as a 0-*supporter*. In a preference profile $\mathcal{P} := \{0,1\}^n$, if the number of 0-supporters and the number of 1-supporters are the same, we say that $\mathcal{P}$ is *balanced*; else we say that it is *unbalanced*.

Unless otherwise noted, we assume that all parties' preferences are predetermined and *public*. We discuss the private-preference case in the appendices, Section 7.

Coin-toss protocol. Consider a protocol Π where n parties jointly decide an outcome between 0 and 1. Such a protocol Π is said to be a coin toss protocol, there is a polynomial-time computable deterministic function, which, given the transcript of the protocol execution, outputs a bit $b \in \{0,1\}$, often said to be the *outcome* of the protocol. For correctness, we require that an honest execution outputs each bit with probability exactly $\frac{1}{2}$ unless all parties have the same preference. More formally, correctness requires that

1. If some parties have differing preferences, in an all-honest execution (when all parties are honest), the probability that the outcome is 0 (or 1) is exactly[8] 1/2.
2. If all parties happen to prefer the same bit $b \in \{0,1\}$, the honest execution should output the preferred bit b with probability 1.

Payoff function. If the protocol's outcome is b, a party who prefers b receive a reward (or payoff) of 1; else it receives a reward (or payoff) of 0. Note that earlier in Section 1, our 1-bit roulette example had a -1 utility (rather than 0) for losing, but the two definitions are in fact equivalent; and for simplicity the remainder of the paper will assume 0 utility for losing.

3.2 Discussions

Trivial case: unanimous preference profile. When everyone has the same preference, we say that the preference profile is *unanimous*; otherwise we say that it is *mixed*. In this case, we do not require that an honest execution produce an unbiased coin, since it makes sense for the outcome to be the bit that is globally preferred. In the remainder of the paper, for the case of public preference: if

[8] Our upper bounds achieve perfect correctness, but our lower bounds in fact extend easily even when allowing negligible correctness failure.

everyone prefers the same bit $b \in \{0,1\}$, we assume that *the protocol simply fixes the outcome to be the universally preferred bit b* regardless of how parties act. In this way, everyone obtains a payoff of 1, and no deviation from the protocol can influence the outcome — therefore all game-theoretic fairness notions we consider are trivially satisfied when the preference profile is unanimous.

On public verifiability. Our definition implies public verifiability of the coin toss's outcome. Anyone who can observe messages sent over the broadcast medium (e.g., a public blockchain) can independently compute the outcome of the protocol. Note that under this definition, the outcome of the protocol is well-defined even when all parties are corrupt. Alternatively, we can define a weaker notion where we do not require such public verifiability — instead we require that honest parties output a bit at the end of the execution, and that they output the same bit (said to be the *outcome* of the execution) with probability 1 even in the presence of an arbitrary (possibly unbounded) adversary that corrupts up to $n-1$ parties. Under this weaker notion, the outcome of an execution is not well-defined when all parties are corrupt. We note that all lower bounds in this paper in fact apply to this weaker notion too (which makes the lower bounds stronger).

3.3 Strong Fairness

We quickly review the classical notion of strong fairness [19]. Roughly speaking, strong fairness requires that the outcome of the coin toss protocol be unbiased even in the presence of an adversary (assuming that parties have mixed preferences). In the definition of strong fairness, we consider a single adversarial coalition that corrupts up to $n-1$ parties.

Definition 1 (Strong fairness [19]). *Let $\mathfrak{A}$ a family of adversaries that corrupt at most $n-1$ parties. An n-party coin toss protocol is said to be strongly fair against the family $\mathfrak{A}$, iff for every adversary $\mathcal{A} \in \mathfrak{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that (as long as not all parties have the same preference) the probability that the outcome is 1 is within $[\frac{1}{2} - \mathsf{negl}(\kappa), \frac{1}{2} + \mathsf{negl}(\kappa)]$ when playing with $\mathcal{A}$.*

4 Maximin Fairness: Feasibilities and Infeasibilities

4.1 Definition of Maximin Fairness

Maximin fairness requires that no honest party should be harmed by any corrupt coalition. In other words, a corrupt coalition should not be able to (non-negligibly) decrease the expected payoff for any honest party relative to an all-honest execution. In maximin fairness, we consider a single adversarial coalition that controls up to $n-1$ parties.

Definition 2 (Maximin fairness). *Let $\mathfrak{A}$ be a family of adversaries that corrupt up to $n-1$ parties; and let $\mathcal{P} \in \{0,1\}^n$ denote any mixed preference profile.*

We say that an n-party coin toss protocol is maximin fair for $\mathcal{P}$ against the family $\mathfrak{A}$, iff for every adversary $\mathcal{A} \in \mathfrak{A}$, there exists some negligible function $\mathsf{negl}(\cdot)$ such that in an execution with the preference profile $\mathcal{P}$ and the adversary $\mathcal{A}$, the expected reward for any honest party is at least $\frac{1}{2} - \mathsf{negl}(\kappa)$. More specifically, we have the following special cases:

- *Computational maximin fairness. If $\mathfrak{A}$ is the family of all non-uniform, probabilistic polynomial-time (henceforth denoted p.p.t.) fail-stop (or malicious resp.) adversaries that can corrupt as many as $n-1$ parties, we say that the protocol is computationally maximin fair for $\mathcal{P}$ against any fail-stop (or malicious resp.) adversaries.*
- *Statistical maximin fairness. If $\mathfrak{A}$ is the family of all fail-stop (or malicious resp.) adversaries (including even computationally unbounded ones) that can corrupt as many as $n-1$ parties, we say that the protocol is statistically maximin fair for $\mathcal{P}$ against any fail-stop (or malicious resp.) adversaries.*
- *Perfect maximin fairness. If a protocol is statistically maximin fair against fail-stop (or malicious resp.) adversaries, and moreover the above definition is satisfied with a choice of 0 for the negligible function, we say that the protocol is perfectly maximin fair for $\mathcal{P}$ against fail-stop (or malicious resp.) adversaries. A perfectly maximin fair protocol does not allow any single honest party to have even negligibly small loss in its expected payoff in comparison with an all-honest execution.*

A straightforward observation is that classical strong fairness (Definition 1) implies maximin fairness:

Fact 1 *If an n-party coin toss protocol Π is strongly fair against a family of adversaries $\mathcal{F}$, then Π is maximin fair against $\mathcal{F}$ for any mixed preference profile $\mathcal{P} \in \{0,1\}^n$.*

Sometimes we also say that a protocol is computationally (or statistically, perfectly resp.) maximin fair for $\mathcal{P}$ against any fail-stop (or malicious resp.) coalition of size K — and this means the most obvious where in the above definitions, the family of adversaries $\mathfrak{A}$ we consider is additionally restricted to corrupting exactly K parties.

Claim. Let $\mathcal{P} \in \{0,1\}^n$ be any mixed preference profile. An n-party coin toss protocol Π satisfies computational (or statistical, perfect resp.) maximin fairness for $\mathcal{P}$ against any fail-stop (or malicious resp.) coalition, iff Π satisfies computational (or statistical, perfect resp.) maximin fairness for $\mathcal{P}$ against any (or malicious resp.) coalition of size exactly $n-1$.

Game theoretic interpretation. If a coin-toss protocol is maximin fair, then the following hold:

1. First, the honest strategy maximizes a player's worst-case expected payoff (even when everyone else is colluding against the player); this explains the name "maximin fairness".

2. Moreover, when playing the honest strategy, a player's worst-case payoff is what it would have gained in an all-honest execution — note that a player's worst-case (expected) payoff obviously cannot be more than its payoff in an all-honest execution.

Equivalence to group maximin fairness. An alternative way to define "no-harm to honest parties" is to require that any corrupt coalition cannot decrease (by more than a negligible amount) the expected overall wealth (i.e., total payoff) of the honest parties. We prove that this notion, called group maximin fairness, is in fact equivalent to maximin fairness in the context of coin toss. We defer the formal definition and proofs to our online full version [18].

4.2 The Case of Amply Divided Preference Profiles

As mentioned, feasibility for 2 parties or multiple parties but honest majority are already implied by existing literature [16, 27]. Henceforth we focus on the case of three or more parties and corrupt majority.

First, we consider amply divided preference profiles, where at least two people prefer 0 and at least two prefer 1 respectively (and hence there must be at least 4 people). It is not too difficult to rule out maximin fairness for amply divided preference profiles, even against fail-stop, computationally bounded adversaries, leading to the following theorem.

Theorem 5 (Maximin fairness: amply divided preference profiles). *For any $n \geq 4$ and for any amply divided preference profile $\mathcal{P} \in \{0,1\}^n$, no n-party coin toss protocol can achieve even computational maximin fairness for $\mathcal{P}$ against even fail-stop adversaries.*

Proof. (sketch.) We show that if there is a maximin fair protocol for any amply mixed preference profile, we can construct a 2-party strongly fair coin toss protocol (and thus violating Cleve's lower bound [19]). The proof follows from a standard partitioning argument: consider two partitions, each containing at least one 0-supporter and at least one 1-supporter. Now, we can view the protocol as a two-party protocol between the two partitions, and by maximin fairness, if either partition aborts, it must not create any non-negligible bias towards either direction. We defer the full proof to our online full version [18].

4.3 The Case for Almost Unanimous Preference Profiles

Possibility of perfect maximin fairness for fail-stop adversaries. First, we show that for fail-stop adversaries, we can achieve perfect maximin-fairness for almost unanimous preference profiles. Without loss of generality, assume that a single party prefers 0 and everyone else prefers 1. The following simple protocol can guarantee perfect maximin fairness:

1. In the first round, the single 0-supporter flips a random coin b and broadcasts b;

2. If the single 0-supporter successfully broadcast a message b, then the outcome is b; else the outcome is 1.

It is not difficult to see that this simple protocol satisfies perfect maximin fairness against fail-stop adversaries: all the 1-supporters do not take any actions and they do not influence the outcome of the protocol. For the single 0-supporter, if it deviates (by aborting), then the outcome will be 1 with probability 1, and all honest parties utility are guaranteed to be 1. Thus we derive the following theorem:

Theorem 6 (Possibility of perfect maximin fairness for almost unanimous preferences and fail-stop adversaries.). *For any $n \geq 3$, any almost unanimous preference profile $\mathcal{P} \in \{0,1\}^n$, there exists an n-party coin toss protocol that achieves perfect maxmin fairness for $\mathcal{P}$ against fail-stop adversaries.*

Impossibility of computational maximin fairness for semi-malicious adversaries. Next, we show that maximin fairness is impossible to achieve for almost unanimous preference profiles against malicious adversaries, even when allowing computational assumptions.

Theorem 7 (Impossibility of maximin fairness for almost unanimous preferences and malicious adversaries). *For $n \geq 3$ and any almost unanimous preference profile $\mathcal{P} \in \{0,1\}^n$, no n-party coin-toss protocol Π can ensure computational maximin fairness for $\mathcal{P}$ against malicious adversaries.*

4.4 Informal Proof Roadmap for Theorem 7

We in fact prove a stronger lower bound than stated in Theorem 7: we show that maximin fairness is impossible for any almost unanimous preference profile (for 3 or more parties), even against *semi-malicious*, polynomially bounded adversaries. In particular, a semi-malicious adversary can 1) choose corrupt parties' random coins arbitrarily upfront, and 2) prematurely abort; but otherwise it follows the honest protocol.

For simplicity, we focus on the case of 3 parties but the proof generalizes directly to more parties. Suppose that the 3 parties are called P_1, P_2 and P_3, and they come with the preferences $1, 0$, and 1 respectively.

We now present an informal proof roadmap, deferring the formal proof to our online full version [18]. We begin by assuming that a maximin fair protocol exists for 3 parties, resisting semi-malicious, computationally bounded adversaries. Our proof will seek to reach a contradiction, effectively showing that the various conditions imposed by maximin fairness cannot co-exist.

Almost All Random Coins of a Lone Semi-Malicious 0-Supporter are Created Equal By a direct application of maximin fairness, if the single 0-supporter is semi-malicious and allowed to program his random coins, then he

should not bias the remaining two parties towards 0. However, perhaps somewhat surprisingly at first sight, we can prove a result that is much stronger, that the single 0-supporter in fact (almost) cannot cause bias towards *either* direction by programming its random coins!

Henceforth, we shall use the notations T_1, T_2, T_3 to denote the three parties' random coins, where T_2 belongs to the single 0-supporter P_2. Consider an honest execution of the protocol conditioned on the fact that the single 0-supporter has its randomness fixed to T_2, and let $f(T_2)$ denote the expected outcome (where the probability is taken over P_1 and P_3's randomness). We prove the following lemma stating that (except for a negligible fraction of choices), all choices of T_2 are equal if P_2 is the lone semi-malicious party.

Lemma 1 (Almost all random coins of a lone semi-malicious P_2 are created equal). *Suppose that the protocol under consideration satisfies computational maximin fairness against semi-malicious adversaries. Then, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that except for* $\mathsf{negl}(\kappa)$ *fraction of T_2's, it must be that* $|f(T_2) - 0.5|$ *is a negligible function in* κ.

Proof. (sketch.) We present a proof sketch: by maximin fairness, we know that for all T_2's, $f(T_2) \geq 0.5 - \mathsf{negl}(\kappa)$. Now, notice that $\mathbf{E}_{T_2} f(T_2) = \frac{1}{2}$ by honest execution, i.e., the expected value of $f(T_2)$ is $\frac{1}{2}$ when averaging over T_2. This means that if there is a non-negligible fraction of T_2's that cause non-negligible bias towards 1, then there must be a non-negligible fraction of T_2's that cause non-negligible bias towards 0 and the latter violates maximin fairness for a semi-malicious P_2.

The Lone-Wolf Condition and Wolf-Minion Conditions Henceforth, our general plan is to show that if the above T_2-equality lemma holds, then the following two conditions, implied by the definition of maximin fairness, cannot co-exist.

- *Lone-wolf condition.* When P_1 (or P_3) is the only fail-stop party, it cannot cause non-negligible bias towards either direction. Such an attack is also called a lone-wolf attack.
- *Wolf-minion condition.* When P_1 and P_2 (or P_2 and P_3) form a fail-stop coalition, they cannot cause non-negligible bias towards 0. In fact we only care about attacks where P_2 is a silent accomplice (called a *minion* [25]) that never aborts but shares information with P_1 (or P_3); and P_1 (or P_3) may abort depending on its view in the execution (called a *wolf* [25]). Such attacks are called wolf-minion attacks.

Note that both conditions above consider only fail-stop adversaries, and in fact in the entire proof the only place we rely on a semi-malicious adversary is in the proof of the aforementioned T_2-equality lemma.

Non-Blackbox Application of Cleve's Lower Bound Conditioned on T_2 Recall that we assume that a maximin fair, 3-party protocol Π exists for

the sake of reaching a contradiction. Now consider an execution of this protocol when P_2's randomness is fixed to T_2, and further, assume that P_2 never aborts and always follows the honest protocol to completion. We now view this 3-party protocol as a 2-party protocol between P_1 and P_3, where P_2's randomness T_2 is public and hard-wired in P_1 and P_3's program — more specifically both P_1 and P_3 would run the 3-party protocol Π, and they each independently simulate the actions of P_2 and compute all messages that P_2 wants to send.

Due to the T_2-equality lemma, if P_1 and P_3 are honest, we know that the expected outcome would be $\frac{1}{2}$ for almost all T_2's. Now, in this 2-party protocol defined by a fixed T_2 (that does not belong to the negligible fraction of bad T_2's), Cleve [19] showed that there must exist a polynomial-time attack by one of the parties, that causes non-negligible bias — but the bias can be either towards 0 or 1. Unfortunately, the direct implication of Cleve's lower bound is not quite so useful for us: it shows that a semi-malicious P_2 can collude with a fail-stop P_1 (or P_3) and cause bias for the remaining honest party, that is P_3 (or P_1) — but unless this bias is towards 0, it does not lend to a contradiction.

Our plan is the following: we will nonetheless apply Cleve's impossibility, but in a non-blackbox manner. First, we will show that for any fixed T_2 (except for a negligible fraction of bad ones), either P_1 or P_3 can bias towards 1 with an aborting attack. Specifically, we define a sequence of adversaries like in Cleve's proof, denoted $\{\mathcal{A}_i^b(1^\kappa, T_2), \mathcal{B}_i^b(1^\kappa, T_2)\}_{i\in[R]}, \cup\{\mathcal{A}_0(1^\kappa, T_2)\}$ where R is the protocol's round complexity. Adversaries $\mathcal{A}_i^b(1^\kappa, T_2)$, $\mathcal{B}_i^b(1^\kappa, T_2)$, and $\mathcal{A}_0(1^\kappa, T_2)$ are defined when P_2's randomness is fixed to T_2:

- Adversary $\mathcal{A}_i^b(1^\kappa, T_2)$:
  - $\mathcal{A}_i^b$ executes the honest protocol on behalf of P_1 and P_2 (whose randomness is fixed to T_2) until the moment right before P_1 is going to broadcast its i-th message.
  - At this moment, $\mathcal{A}_i^b$ computes α_i, that is, imagine that P_3 aborted right after sending its $(i-1)$-th message, what would be the outcome of parties P_1 and P_2.
  - If $\alpha_i = b$, then P_1 aborts after sending the i-th message; else P_1 aborts right now without sending the i-th message.
- Adversary $\mathcal{B}_i^b(1^\kappa, T_2)$: The definition is symmetric to that of $\mathcal{A}_i^b(1^\kappa, T_2)$ but now P_3 is the fail-stop party.
- Adversary $\mathcal{A}_0(1^\kappa, T_2)$: P_1 aborts upfront prior to speaking at all.

Cleve [19] showed that one of these above adversaries must be able to cause non-negligible bias towards either 0 or 1. However, due to the requirement of maximin fairness, we may conclude that the bias must be towards 1 except for a negligible fraction of the T_2's. Suppose this is not the case, i.e., the bias is towards 0 for a non-negligible fraction of the T_2's — then we could easily construct an attack (for the 3-party protocol) where a semi-malicious P_2 colluding with a fail-stop P_1 (or P_3) can bias the remaining party towards 0 — in fact, in our formal proof later, we show that such an attack is even possible with a fail-stop P_1 and a silent accomplice P_2 who just shares information with P_1 but would

otherwise follow the protocol honestly (i.e., a wolf-minion attack). Proving this stronger statement would require a little more effort — but looking forward, later we would like to rule out CSP fairness for even fail-stop adversaries. There we have a similar agenda: 1) reprove the T_2-equality lemma but for fail-stop adversaries and CSP fairness, and 2) show that under the T_2-equality lemma, the lone-wolf condition and the wolf-minion condition cannot co-exist. Thus in our formal proof later we will actually rely on a wolf-minion (fail-stop) attack to rule out the 0-bias attack.

Averaging over T_2: A Wolf-Minion Attack with Benign Bias Next, we consider the above adversaries but now averaging over T_2. In other words, let $\overline{\mathcal{A}}_i^b(1^\kappa)$ be the following attacker: choose a random T_2; now consider the protocol execution with P_2's randomness fixed to T_2 and with the adversary $\mathcal{A}_i^b(1^\kappa, T_2)$. $\overline{\mathcal{B}}_i^b(1^\kappa)$ and $\overline{\mathcal{A}}_0(1^\kappa)$ are similarly defined by averaging over T_2.

Now, we prove that among these adversaries $\{\overline{\mathcal{A}}_i^b(1^\kappa), \overline{\mathcal{B}}_i^b(1^\kappa)\}_{i\in[R]}$ and $\overline{\mathcal{A}}_0(1^\kappa)$, one of them must be able to bias the remaining party, either P_1 or P_3, towards 1. This proof follows in a somewhat standard manner from an averaging argument and we defer the details to the appendices. Note that reflecting in the 3-party protocol, this corresponds to a wolf-minion attack that creates benign bias: P_1 (or P_3) acts as a fail-stop wolf, and P_2 acts as a silent accomplice (i.e., the minion) that follows the honest protocol to completion but shares information with P_1 (or P_3). Although this wolf-minion is able to create bias, the bias is benign and does not violate the definition of maximin fairness. Thus to reach a contradiction, it still remains to show an attack that creates harmful bias.

Applying the Lone-Wolf Condition: A Wolf-Minion Attack with Harmful Bias We now argue that if there is a wolf-minion attack that creates benign bias, there must be one that creates harmful bias, assuming that the lone-wolf condition holds. To show this, we consider the adversary that flips the decisions (to abort in the present or next round) of the benign wolf-minion attack. Without loss of generality, assume that $\overline{\mathcal{A}}_i^1$ is the successful wolf-minion attack that creates non-negligible bias towards 1. We now consider $\overline{\mathcal{A}}_i^0$ which flips $\overline{\mathcal{A}}_i^1$'s decision whether to abort in round i or $i+1$, and we argue that $\overline{\mathcal{A}}_i^0$ must create non-negligible bias towards 0. At a very high level, the proof will show that the lone-wolf condition acts like a balancing condition.

Let Q be the set of sample paths (defined by choices of T_1, T_2, and T_3) over which $\overline{\mathcal{A}}_i^1$ decides to abort in round i, and let $\overline{Q}$ be the remaining sample paths. Now, consider a hybrid adversary that takes $\overline{\mathcal{A}}_i^1$'s decisions on Q and takes $\overline{\mathcal{A}}_i^0$'s decisions on $\overline{Q}$: in other words, P_1 basically always aborts in round i! Due to the lone-wolf condition, whatever average bias towards 1 $\overline{\mathcal{A}}_i^1$ has on Q, $\overline{\mathcal{A}}_i^0$ must create almost the same bias towards 0 on $\overline{Q}$. By a symmetric argument and considering a lone wolf P_1 that always aborts in round $i+1$, whatever average bias towards 1 $\overline{\mathcal{A}}_i^1$ has on $\overline{Q}$, $\overline{\mathcal{A}}_i^0$ must create almost the same bias towards 0 on

Q. With this, it is not difficult to see that $\overline{\mathcal{A}}_i^0$ can bias towards 0 (almost) as well as $\overline{\mathcal{A}}_i^1$ can bias towards 1.

5 Cooperative-Strategy-Proof Fairness

5.1 Definition of Cooperative-Strategy-Proof Fairness

In a cooperative strategy, a corrupt coalition deviates from the honest protocol in an attempt to improve the coalition's overall wealth (i.e., the total reward). Cooperative strategies naturally arise in contexts where a corrupt coalition is allowed to have binding side contracts that allow the coalition to redistribute (e.g., equally) the overall wealth among its members. If a protocol is cooperative-strategy-proof fair (or CSP-fair), it intuitively means that any corrupt coalition should not be able to improve its overall wealth by more than negligible amounts (if the remaining parties are faithfully following the honest protocol).

Definition 3 (Cooperative-strategy-proof fairness or CSP-fairness). *Let $\mathfrak{A}$ be a family of adversaries that corrupt up to $n-1$ parties and let $\mathcal{P} \in \{0,1\}^n$ denote any mixed preference profile. We say that an n-party coin toss protocol is cooperative-strategy-proof fair (or CSP-fair) for $\mathcal{P}$ and against the family $\mathfrak{A}$, iff for any adversary $\mathcal{A} \in \mathfrak{A}$, there exists some negligible function $\mathsf{negl}(\cdot)$, such that in an execution with the preference profile $\mathcal{P}$ and the adversary $\mathcal{A}$, the expected total reward for the set of corrupt parties (denoted C) is at most $\sigma(C) + \mathsf{negl}(\kappa)$ where $\sigma(C)$ denotes the expected total reward for all nodes in C in an all-honest execution.*

Similar as before, now depending on the family $\mathfrak{A}$ of adversaries that we are concerned about, we can define computational, statistical, or perfect notions for cooperative-strategy-proof fairness, and for fail-stop, semi-malicious, or malicious adversaries respectively. We omit the detailed definitions for conciseness.

Remark 1 (The case of a global coalition for CSP-fairness). Unless otherwise noted, the definition of CSP-fairness considers coalitions of size up to $n-1$. One could alternatively define a variant of CSP-fairness where the corrupt coalition can contain up to n parties, i.e., CSP-fairness is desired even against a global coalition where everyone is corrupt. For any *balanced* preference profile, this variant is equivalent to the definition where not all can be corrupt since the global coalition is indifferent to either outcome. For any *unbalanced* preference profile, this variant where all can be corrupt is a stronger notion — in fact, one could easily rule out feasibility against (even computationally bounded) semi-malicious adversaries due to the following argument. By correctness, there must exist some joint randomness $\boldsymbol{\rho}$ of all parties, such that an honest execution fixing the randomness to $\boldsymbol{\rho}$ would lead to the outcome that is preferred by the global coalition. Now a semi-malicious adversary can receive this $\boldsymbol{\rho}$ as advice and program the parties' joint randomness to $\boldsymbol{\rho}$. Interestingly, however, for fail-stop adversaries, we will show the feasibility of perfect CSP-fairness even when all parties can be corrupt (see Corollary 1).

For any *balanced* preference profile, if the corrupt coalition gains in terms of overall wealth (i.e., total payoff) then honest overall wealth must be harmed (relative to an honest execution in both cases). Therefore, CSP-fairness is equivalent to maximin fairness for balanced preference profiles. The following fact is therefore straightforward:

Fact 2 (Equivalence of maximin fairness and CSP fairness for balanced preference profiles) *Let $\mathfrak{A}$ denote a family of adversaries that corrupt up to $n-1$ parties and let $\mathcal{P} \in \{0,1\}^n$ denote any balanced profile. Then, an n-party coin toss protocol Π is maximin fair for $\mathcal{P}$ against the family $\mathfrak{A}$ iff Π is CSP-fair for $\mathcal{P}$ against the family $\mathfrak{A}$.*

For *unbalanced* preference profiles, however, the two notions are not equivalent (and this will become obvious later in the paper).

As mentioned, for two parties, all our fairness notions equate to Blum's weak fairness notion [16], and therefore the results stated in our online full version [18] directly apply to CSP fairness too. In the remainder of this section, we focus on three or more parties.

5.2 Almost Unanimous Preference Profile

Recall that we consider 3 or more parties, i.e., $n \geq 3$.

Possibility of perfect CSP-fairness against semi-malicious adversaries. First, we show that for almost unanimous preference profiles and any $n \geq 3$, perfect CSP-fairness is possible against any coalition of size up to $n-1$.

Let $P_0, \ldots, P_{n-1}$ denote the $n \geq 3$ players. Without loss of generality, suppose that P_0 is the single 0-supporter (i.e., prefers 0), and everyone else prefers 1 (all other cases are equivalent by flipping the bit and renumbering players). Consider the following simple protocol denoted Π_{csp}.

1. In the first round, every party i where $i \in [0, 1, \ldots, n-1]$ locally tosses a random coin b_i. Further, the single 0-supporter P_0 reveals its coin b_0.
2. In the second round, every 1-supporter (i.e., P_i where $i \neq 0$) reveals coin b_i.
3. The outcome of the protocol is defined as follows: if any 1-supporter aborted without revealing its bit, output 0. Else, output the XOR of all bits that have been revealed by the parties — note that if P_0 aborted without revealing its bit b_0, then we simply do not include b_0 in the XOR.

It is straightforward that under an honest execution, the expected outcome is $\frac{1}{2}$.

Theorem 8 (Possibility of perfect CSP-fairness against semi-malicious corruptions for almost unanimous preference profiles). *For any $n \geq 3$, there is an n-party coin toss protocol that achieves perfect CSP-fairness for any almost unanimous preference profile $\mathcal{P} \in \{0,1\}^n$ against the family of all semi-malicious adversaries that control at most $n-1$ parties.*

Proof. We analyze the aforementioned protocol Π_{csp} by considering the following cases:

1. P_0 **is the lone corrupt party**. In this case, all parties who prefer 1 are honest, and since P_0 makes its decision to abort prior to seeing the remaining parties' random bits, equivalently, we can think of the remaining parties flip their random coins after P_0 makes its decision whether to abort. Thus, regardless of P_0's strategy, the expected outcome must be $\frac{1}{2}$.
2. P_0 **and a single 1-supporter are corrupt.** In this case, the definition of CSP-fairness is trivially satisfied since the corrupt coalition would obtain a payoff of exactly 1 no matter what the outcome of the protocol is.
3. P_0 **is honest and one or more 1-supporters are corrupt.** Let $\mathbf{b} := (b_0, \ldots, b_{n-1})$ denote the random coin tosses of all the parties. For semi-malicious corruption, we can imagine that each party P_i chooses b_i and other randomness related to aborting decisions upfront prior to protocol start — honest parties sample them at random and corrupt parties choose the random strings arbitrarily. Let C denote the corrupt coalition and let $-C$ denote its complement. We consider an alternative adversary $\mathcal{B}$ that just receives $\mathbf{b}^C := \{b_i\}_{i \in C}$ as advice but all corrupt parties follow the protocol to the end — note that such a $\mathcal{B}$ needs to consume only $\mathbf{b}^C$ and no additional randomness. For any fixed $\mathbf{b}^C$, and for any fixed $\mathbf{b}^{-C}$, if playing with the adversary $\mathcal{B}$ who never aborts, the outcome is 0, then playing with any adversary $\mathcal{A}$ (who might abort), the outcome cannot be 1. Thus for every $(\mathbf{b}^C, \mathbf{b}^{-C})$, no adversary $\mathcal{A}$ can obtain a higher outcome than $\mathcal{B}$. The proof follows by seeing that for $\mathcal{B}$ and for any fixed $\mathbf{b}^C$, the expected outcome (averaging over honest parties' random coin flips) is $\frac{1}{2}$.
4. P_0 **and at least two 1-supporters are corrupt.** In this case it must be that $n \geq 4$ since if $n = 3$ all parties would be corrupt. Similar to the above case, here we can argue that for every fixed $(\mathbf{b}^C, \mathbf{b}^{-C})$ and P_0's decision whether to abort, the adversary $\mathcal{B}$ such that all other corrupt corrupt (besides P_0) execute to the end makes the outcome at least as high as any other adversary $\mathcal{A}$. Additionally, for $\mathcal{B}$ and for any fixed $\mathbf{b}^C$ and P_0's decision whether to abort, the expected outcome (averaging over honest parties' random coin flips) is $\frac{1}{2}$.

Corollary 1. *There is an 3-party coin toss protocol that achieves perfect CSP-fairness for any mixed preference profile against the family of all semi-malicious adversaries that control at most $n-1$ parties.*

Proof. Note that for 3 parties, any mixed preference profile must be almost unanimous. The corollary now follows from Theorem 8.

We observe that for fail-stop adversaries, a variant of the aforementioned protocol Π_{csp} actually achieves perfect CSP-fairness even when all parties can be corrupt: Suppose that only parties in $\{P_1, \ldots, P_{n-1}\}$ flip a random coin and publish the coin; and P_0 does nothing. If any of these parties abort, the outcome is defined to be 0; else the outcome is defined to be the XOR of all published coins. We thus have the following corollary:

Corollary 2. *For any $n \geq 3$, there is an n-party coin toss protocol that achieves perfect CSP-fairness for any almost unanimous preference profile $\mathcal{P} \in \{0,1\}^n$ against the family of all fail-stop adversaries that control up to n parties.*

Proof. If no 1-supporter is corrupt, then obviously the expected outcome is $\frac{1}{2}$. If at least one 1-supporter is corrupt, then for every choice of the joint randomness of all 1-supporters, having any 1-supporter abort does no better for the adversary than having no 1-supporter abort.

Possibility of computational CSP-fairness against malicious adversaries. It is also easy to see that for three or more parties, *statistical* CSP fairness is impossible against malicious adversaries much as the 2-party case [33,44]. Therefore for malicious adversaries we have to make computational assumptions.

For conceptual simplicity, we first describe our protocol assuming an idealized commitment scheme — in our online full version [18], we describe how to dispense with this idealized primitive and realize it from concurrent non-malleable commitments that can be constructed one-way permutations. For the time being, imagine that there is a special trusted party called $\mathcal{F}_{\text{idealcomm}}$ that has the following interface:

- In the first round (i.e., the commitment round), if $\mathcal{F}_{\text{idealcomm}}$ receives (`commit` b) from some party i, it tells everyone (`committed`, i).
- In any of the subsequent rounds (i.e., the opening rounds), if $\mathcal{F}_{\text{idealcomm}}$ receives `open` from any party i who has committed b_i in the first round, it tells everyone (`open`, i, b_i).

We can now upgrade our semi-malicious protocol earlier to resist even malicious adversaries (w.l.o.g. assume that there is a single 0-supporter and everyone else is a 1-supporter):

1. In round 0, everyone commits a bit to $\mathcal{F}_{\text{idealcomm}}$;
2. In round 1, the single 0-supporter opens its commitment;
3. In round 2, everyone else opens;
4. If any 1-supporter aborted, the outcome is 0; else the outcome is the XOR of all bits that have been opened.

Since the commitment round basically forces corrupt parties to commit to their randomness upfront; it is easy to see that this new protocol is CSP-fair against malicious adversaries (for the same reason why the earlier protocol is CSP-fair against semi-honest adversaries). Note that CSP fairness holds even for unbounded adversaries assuming the $\mathcal{F}_{\text{idealcomm}}$ ideal functionality; but in our online full version [18], we show how to remove the $\mathcal{F}_{\text{idealcomm}}$ and replace it with concurrent non-malleable commitments [42], the resulting protocol would secure only against computationally bounded adversaries as stated in the following theorem.

Theorem 9 (Computational CSP fairness against malicious adversaries). *Assume that one-way permutations exist, then for any $n \geq 3$, there exists an*

n-party protocol that achieves computational CSP fairness for any almost unanimous preference profile $\mathcal{P} \in \{0,1\}^n$ against malicious coalitions of size up to $n-1$.

The proof is deferred to our online full version [18].

5.3 Amply Divided Preference Profile

For $n = 3$, any mixed preference profile must be almost unanimous. For $n \geq 4$, we need to consider amply divided preference profiles: i.e., at least two parties prefer 0 and at least two parties prefer 1. We now show a strong impossibility for mixed preference profiles, that is, for any mixed preference profile $\mathcal{P}$, no n-party coin toss can achieve even computational CSP-fairness for $\mathcal{P}$ against even fail-stop adversaries.

We note that for the special case of amply divided and *balanced* preference profiles, the impossibility for CSP fairness is already implied by the impossibility of maximin fairness for the same preference profiles (Theorem 5) — recall that the two notions are equivalent for balanced preference profiles. However, this observation does not rule out the feasibility of CSP fairness for *unbalanced* and amply divided preference profiles. Thus the following theorem is non-trivial even in light of Theorem 5.

Theorem 10 (Impossibility of CSP-fairness for $n \geq 4$). *Let $n \geq 4$, and let $\mathcal{P} \in \{0,1\}^n$ be any amply divided preference profile. Then, no n-party coin-toss protocol can achieve even computational CSP-fairness for $\mathcal{P}$, against even fail-stop adversaries.*

Proof roadmap. Although for *balanced* and amply mixed preference profiles, the infeasibility of CSP fairness is already implied by the infeasibility of maximin fairness for the same profiles (since the two notions are equivalent for balanced preference profiles), here we would like to prove impossibility for *any* amply mixed preference profile, even *unbalanced* ones. At a very high level, our approach is to group the parties into three partitions called P_1, P_2, and P_3, such that we can view the execution as a 3-party protocol. This partitioning is carefully crafted such that the definition of CSP fairness would imply the T_2-equality lemma, the lone-wolf condition, and the wolf-minion conditions like in the impossibility proof for maximin fairness — and if this is the case, the same proof would apply and rule out CSP fairness.

Among these conditions, the T_2-equality lemma is the most challenging to prove. Specifically, earlier we relied on maximin fairness against a *semi-malicious* P_2 to prove the T_2-equality lemma; and here would like to prove the same lemma for CSP fairness but now against a *fail-stop* adversary[9]. This seems almost counter-intuitive at first sight since at the surface, the T_2-equality lemma is

[9] Note that the T_2-equality lemma does not even hold for maximin fairness against *fail-stop* adversaries since we have an explicit construction for almost pure preference profiles and fail-stop.

stating that *if a semi-malicious adversary were to program T_2 to specific strings, almost for all such strings it would not help*. But now how can we prove it by relying on CSP fairness against only *fail-stop* adversaries? In our formal proof later, we will show that for any two neighboring T_2 and T_2' (except for a negligibly small bad fraction), it must be that $|f(T_2) - f(T_2')| \leq \mathsf{negl}(\kappa)$, where T_2 and T_2' are said to be neighboring iff they differ only in one party's contribution of random coins, and $f(T_2)$ is defined similarly as before, i.e., the expected outcome of an honest execution conditioned on P_2's randomness being fixed to T_2. Now if we can show this, we can then show, through a hybrid argument, that $|f(T_2) - f(T_2')| \leq \mathsf{negl}(\kappa)$ for any T_2 and T_2' (except for a negligibly small bad fraction), and this would complete the proof.

Thus the challenge is to show $|f(T_2) - f(T_2')| \leq \mathsf{negl}(\kappa)$ for almost all *neighboring* T_2 and T_2' pairs. To do this, suppose that T_2 and T_2' differ in the i-th player's contribution where $i \in P_2$ — our intuition is to compare an honest execution involving T_2 with the execution where the i-th player aborts upfront (and P_2's randomness still fixed to T_2). Let $g^i(T_2)$ denote the expected outcome in the latter execution. Through a somewhat non-trivial argument, we will prove that for almost all T_2s, it must be that $|f(T_2) - g^i(T_2)| \leq \mathsf{negl}(\kappa)$ — otherwise we can construct a *fail-stop* adversary in control of P_2, and this adversary, upon generating an honest random T_2, emulates polynomially many honest executions conditioned on T_2 to estimate $f(T_2)$ and $g^i(T_2)$ respectively, and informed by the estimates, decide to either have i abort upfront or not. We prove that such an adversary can cause non-negligible bias that improves P_2's overall wealth.

Similarly, for T_2' that is almost identical as T_2 but differing in the i-th coordinate, we also have that $|f(T_2') - g^i(T_2')| \leq \mathsf{negl}(\kappa)$. Finally, the proof follows by observing that, if the i-th party aborts upfront, then its random coins do not affect the expected outcome of the execution, i.e., $g^i(T_2) = g^i(T_2')$.

We defer the full proof of this theorem to our online full version.

6 Fairness by Strong Nash Equilibrium

6.1 Definition of Strong Nash Equilibrium (SNE)

Strong Nash Equilibrium (SNE) requires that no coalition, corrupting up to n parties, can noticeably (i.e., non-negligibly) increase the payoff of all members of the coalition. SNE is weaker than the earlier CSP notion since the former only needs to resist a subset of the coalition strategies that latter must resist — CSP must not only defend against coalition strategies that benefit all of its members, but also defend against strategies that benefit coalition members on average[10]. More formally, we define SNE-fairness below.

Definition 4 (Strong Nash Equilibrium or SNE-fairness). *Let $\mathfrak{A}$ be a family of adversaries that corrupt up to n parties and let $\mathcal{P} \in \{0,1\}^n$ be any*

[10] Since SNE only needs to defend against unanimous coalitions by Fact 3, for any mixed preference profile we in fact only need to consider coalitions of size $n-1$ rather than n.

mixed preference profile. We say that an n-party coin toss protocol is SNE-fair for $\mathcal{P}$ and against the family of adversaries $\mathfrak{A}$ iff for any $\mathcal{A} \in \mathfrak{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$, such that in an execution with the preference profile $\mathcal{P}$ and the adversary $\mathcal{A}$, there is at least one corrupt party whose expected payoff is less than $\frac{1}{2} + \mathsf{negl}(\kappa)$.

Note that the definition of SNE-fairness requires that the notion be satisfied *even when all parties are corrupt*. Similar as before, depending on the family $\mathfrak{A}$ of adversaries that we are concerned about, we can define computational, statistical, or perfect notions for SNE-fairness, and for fail-stop, semi-malicious, or malicious adversaries respectively. We omit the detailed definitions for conciseness.

A coalition of parties is said to be *unanimous* iff every party in the coalition prefers the same bit.

Fact 3 *Let $\mathfrak{A}$ be a family of adversaries corrupting up to n parties and let $\mathfrak{A}' \subset \mathfrak{A}$ be the (maximal) subset of $\mathfrak{A}$ that corrupts only unanimous coalitions*[11]*. Let $\mathcal{P} \in \{0,1\}^n$ be any mixed preference profile. Then, an n-party coin toss protocol Π is CSP-fair for $\mathcal{P}$ against the family $\mathfrak{A}'$ iff Π is SNE-fair for $\mathcal{P}$ against the family $\mathfrak{A}$.*

Proof. For any adversary $\mathcal{A} \in \mathfrak{A}$ that corrupts a coalition that has mixed preferences, if the coalition memebers that prefer 0 have expected payoff more than $\frac{1}{2}$, then those who prefer 1 must have payoff at most $\frac{1}{2}$ — thus SNE-fairness is trivially satisfied for mixed coalitions. We therefore conclude that a protocol Π to be SNE-fair for $\mathcal{P}$ against $\mathfrak{A}$, if and only if Π is SNE-fair for $\mathcal{P}$ against those adversaries in $\mathfrak{A}$ that control unanimous coalitions — and this latter notion is equivalent to CSP-fair for unanimous coalitions, by observing the following: since $\mathcal{P}$ is mixed, any adversary in $\mathfrak{A}$ that controls unanimous coalitions corrupts only up to $n-1$ parties (recall that the definition of CSP-fair considers adversaries that corrupts upto $n-1$ parties).

6.2 Feasibility Results for SNE Fairness

We show that for any $n \geq 2$, there is an n-party coin toss protocol that is computationally SNE-fair for any mixed preference profile $\mathcal{P} \in \{0,1\}^n$ against even malicious adversaries; further, there is an n-party coin toss protocol that is perfectly SNE-fair for any mixed preference profile $\mathcal{P} \in \{0,1\}^n$ against semi-malicious adversaries. On the other hand, the impossibility of statistical SNE fairness against malicious adversaries is implied in a straightforward fashion by known lower bounds [33, 44].

Achieving perfect SNE-fairness against semi-malicious adversaries. Let $n \geq 3$ and let $\mathcal{P} \in \{0,1\}^n$ be a mixed preference profile. We can consider a simple

[11] Recall that we assume that the choice of corrupt parties is hard-wired in an adversary's algorithm.

dueling protocol: pick two people with opposing preferences (i.e., the ones with the smallest party identifiers) and have them play the simple 2-party protocol: each party picks a random bit upfront and both broadcast their bit in the first round. Normally the outcome is the XOR of the two bits but if one party aborts, the outcome is the other party's preference.

Theorem 11 (Perfect SNE-fairness against semi-malicious adversaries). *For any $n \geq 2$, there is an n-party coin toss protocol that is perfectly SNE-fair for any mixed preference profile $\mathcal{P} \in \{0,1\}^n$ against semi-malicious adversaries.*

Proof. By Fact 3, we only need to resist unanimous coalitions. Thus for the two parties selected to duel with opposing preferences, one of them must be honest. Further, recall that a semi-malicious adversary must select its random coins upfront without seeing any protocol message, and henceforth the only attack it can perform is aborting. Now in the 2-party protocol, for any choice of randomness of the 2 dueling parties, if the corrupt party aborts, it does no better than playing honestly till completion.

Achieving computational SNE-fairness against malicious adversaries. The above protocol can be made secure against malicious adversaries using a cryptographic commitment scheme. The only change needed is that when the selected two parties duel, one of them (denoted P) commits to a bit in Phase 0, then the other party (denoted P') sends its bit in Phase 1, and finally P opens its commitment.

Theorem 12 (Computational SNE-fairness against malicious adversaries). *For any $n \geq 2$, there is an n-party coin toss protocol that achieves computational SNE-fairness for any mixed preference profile $\mathcal{P} \in \{0,1\}^n$ against malicious adversaries.*

Proof. Consider the dueling protocol Π_{duel}. By Fact 3, it suffices to prove that any unanimous coalition cannot non-negligibly improve the coalition's total reward. Notice that any unanimous coalition controls at most one party in the two parties selected to duel. By maximin fairness of the 2-party protocol (which we argue in our online full version [18]), if one of the dueling parties deviates, the deviating party cannot improve its expected payoff by more than a negligible amount.

7 The Case of Private Preference Profiles

Here we consider the case of private preference profiles, where each party's preference is private information only known to the party. In other words, we consider *private preference* coin toss protocols, where each party's preference is a private input, instead of public information. Clearly, this is a more challenging setting for achieving fairness. For example, a malicious party may lie about his preference or abort without revealing his preference. Indeed, as we shall see, we lose some feasibility results in the private preference setting.

Recall that in the public preference setting, coin toss protocols and fairness can be naturally defined with respect to a preference profile $\mathcal{P}$. However, this is not the case for private preference. Thus, we only consider (universal) n-party private preference coin toss protocols that are defined for every preference profiles $\mathcal{P} \in \{0,1\}^n$. All three fairness notions can be naturally defined for such protocols. Below we only state the definition of maximin fairness in the private preference setting formally for succinctness. The other two notions can be defined analogously.

Definition 5 (Maximin fairness). *Let $\mathfrak{A}$ be a family of adversaries that corrupt up to $n-1$ parties. We say that an n-party private preference coin toss protocol is private maximin fair against the family $\mathfrak{A}$, iff for every adversary $\mathcal{A} \in \mathfrak{A}$, there exists some negligible function $\mathsf{negl}(\cdot)$ such that for every mixed preference profile $\mathcal{P} \in \{0,1\}^n$, in an execution with the preference profile $\mathcal{P}$ and the adversary $\mathcal{A}$, the expected reward for any honest party is at least $\frac{1}{2} - \mathsf{negl}(\kappa)$. For unanimous preference profiles, the execution should output the common preference with probability* 1.

We proceed to discuss the feasibility and impossibility of fair coin toss for private preference protocols. As this is harder to achieve, all impossibility results in the public preference setting trivially hold here, and it suffices to investigate cases that are feasible in the public preference setting.

SNE-fairness. Recall that even in the public preference setting, we can only achieve general feasibility result for the notion of SNE-fairness, where computational SNE-fairness against malicious adversary and statistical SNE-fairness against semi-malicious adversary are feasible for any $n \geq 2$ parties (whereas maximum and CSP-fairness are impossible for $n \geq 4$ even against fail stop adversary). In the private preference setting, we show that SNE-fairness against malicious adversary becomes impossible for $n \geq 3$ parties, whereas SNE-fairness against semi-malicious adversary remain feasible. Intuitively, the reason for the impossibility is that a malicious adversary may lie about his preference.

Theorem 13 (Impossibility of SNE-fairness against malicious adversary). *For any $n \geq 3$, no n-party private preference coin-toss protocol can achieve even computational SNE-fairness against malicious adversaries.*

Proof. (sketch) We focus on the three-party case and discuss how to handle general $n \geq 4$ parties at the end of the proof. At a high level, the proof for the three-party case relies on the same argument as that of Theorem 7 for maximin-fairness. Recall that in the proof of Theorem 7, we consider preference profile $\mathcal{P} = (1,0,1)$. We show that maximin-fairness implies T_2-equality lemma (Lemma 1) and the lone-wolf and wolf-minion conditions. Then we use these properties to derive a contradiction by constructing an adversary that breaks the wolf-minion condition. Here, we follow the same strategy to consider preference profile $\mathcal{P} = (1,0,1)$. It suffices to show that private SNE-fairness implies the same set of properties, and a contradiction can be derived in the same way.

Let Π be a three-party private preference coin toss protocol. Recall that we use the notation $T_1, T_2, T_3 \in \{0,1\}^{\ell(\kappa)}$ to denote the randomness of P_1, P_2 and P_3, respectively, and $f(T_2)$ to denote the expected outcome when P_2 uses the randomness T_2 whereas P_1 and P_3 executed the protocol honestly (when the preference profile is $\mathcal{P} = (1,0,1)$).

It is not hard to see that Lemma 1 follows. Please see our online full version [18]. Thus, T_2-equality lemma is implied by private SNE-fairness as well. It remains to check the lone-wolf and wolf-minion conditions.

For the lone-wolf conditions, it may seem that SNE-fairness only implies that P_1 (or P_3) cannot cause non-negligible (in κ) bias towards 1 by a fail-stop attack. This is the place that an adversary can take the advantage of private preference. Suppose there P_1 can cause non-negligible bias towards 0 by a fail-stop attack when the preference profile is $(1,0,1)$. Consider the case that the preference profile is $(0,0,1)$. An malicious P_1 (with preference 0) can participate the protocol with a pretended preference 1 and perform the fail-stop attack to cause bias toward 0 to violate fairness. Thus, a fail stop P_1 (or P_3) cannot cause non-negligible bias towards either direction.

Recall that the wolf-minion condition says that when P_1 and P_2 (or P_2 and P_3) form a fail-stop coalition, they cannot cause a non-negligible (in κ) bias towards 0. Suppose this is not the case, e.g., a fail-stop coalition P_1 and P_2 can cause a non-negligible bias towards 0. We show that SNE-fairness can be violated when the preference profile is $(0,0,1)$. Indeed, in this case, an malicious adversary corrupting P_1 and P_2 can pretend the preference of P_1 is 1 and use the assumed fail-stop attack to cause a non-negligible bias towards 0, which violates SNE-fairness.

The above shows that for three-party protocols, the properties needed in the proof of Theorem 7 are implied by private SNE-fairness. A contradiction can then be derived by the same arguments as in Theorem 7, which proves the impossibility.

Finally, for general $n \geq 4$ parties, we can use the standard trick to group $P_4, \ldots, P_n$ together with P_2 to form a supernode of 0-supporters. This effectively reduce the number of parties to 3 and the same argument can be applied to show impossibility.

Theorem 14 (Perfect private SNE-fairness against semi-malicious adversaries). *For any $n \geq 2$, there is an n-party private preference coin toss protocol that is perfectly private SNE-fair against semi-malicious adversaries.*

Proof. We simply modify the public preference duelling protocol by first asking all parties to reveal their private preference. If any parties abort, we ignore them. For the remaining non-aborting parties, we proceed with the dueling protocol as in the public preference setting. Note that since we only consider semi-malicious adversaries, the revealed preferences must be the true preferences.

By Fact 3 (which can be verified to hold in the private preference setting with the same argument) , we only need to resist unanimous coalitions. Hence, all aborting parties must share the same preference as their non-aborting coalition

(if any), who do not gain any advantage by the fairness of the public preference protocol. If all non-aborting parties are honest, then correctness of the honest execution also implies that the aborting parties do not gain any advantage.

Note that Theorem 14 is proved by a protocol that first asks all parties to reveal their private preference and then executes a public preference protocol among the non-aborting parties, and intuitively, this works since the semi-malicious can only reveal their true preferences. However, while this intuition turns out to be true for SNE-fairness and maximin fairness (which we discuss later), it can be subtle for CSP-fairness since the adversary still has the advantage of aborting before revealing his preference. We discuss this next.

CSP-fairness. Recall that in the public preference setting, Corollary 2 says that for $n \geq 3$, there exists an n-party coin toss protocol that achieves perfect CSP-fairness for any almost unanimous preference profile against all fail-stop adversaries that can control up to n parties. In particular, there exists a three-party perfect CSP-fair protocol against fail-stop adversaries who may corrupt all three parties[12]. Interesting, this becomes impossible in the private preference setting.

Theorem 15 (Impossibility of CSP-fairness against fail-stop all-corruption adversary). *No three-party private preference coin-toss protocol can achieve computational CSP-fairness against fail-stop adversaries that can corrupt up to three parties.*

Proof. (sketch) For the sake of contradiction, suppose Π is a three-party private preference coin-toss protocol that achieve the claimed fairness. Let us consider a scenario where P_3 always abort at the beginning, and P_1 and P_2 has preference 0 and 1, respectively. Note that suppose P_1 and P_2 execute the protocol honestly, the outcome need to be unbiased: Suppose the outcome is biased towards b and the private preference of P_3 is also b, then the CSP-fairness is violated.

Thus, in this scenario where P_3 is aborting, honest P_1 and P_2 execute a two-party protocol and produce an unbiased outcome. We can apply Cleve's lower bound argument to show the existence of a fail-stop adversary P_a that can bias the outcome non-negligibly towards b, for some $a \in \{1, 2\}$ and $b \in \{0, 1\}$. Now, suppose the private preference of P_3 is b, and consider an adversary $\mathcal{A}$ that corrupts all three parties and does the following: (i) $\mathcal{A}$ lets P_3 aborts at the beginning, and (ii) $\mathcal{A}$ let P_a to perform the fail-stop attack to cause non-negligible bias of the outcome towards b. This violates CSP-fairness since the total utility of the corrupted parties is increased by a non-negligible amount.

On the positive side, we observe that Corollary 1 extends to the private preference setting.

[12] We focus on the three-party case here since the case of four or more parties are impossible in the private preference setting due to the existence of amply divided preference profiles for four or more parties.

Theorem 16. *There is an 3-party private preference coin toss protocol that achieves perfect CSP-fairness against the family of all semi-malicious adversaries that control at most 2 parties.*

Proof. (sketch) We follow the same strategy to first ask each party reveal his preference, and then let the non-aborting parties to execute a fair public preference protocol. Specifically, if no party aborts, then we run the three-party CSP-fair protocol in Corollary 1. If one party aborts and the remaining two parties have the same preference, then they output their preference. If one party aborts and the remaining two parties have different preferences, then they execute the dueling protocol. If two parties abort, then the remaining party simply output his preference. It is not hard to see by inspection that private CSP-fairness holds in all cases.

Maximin fairness. We end this section with a brief discussion on the maximin fairness for the private preference protocols. Note that the only interesting question is whether Theorem 6, which states the existence of perfect maximin fair coin toss protocol against fail-stop adversaries, extends to the private preference setting. Now, observe that the definition of maximin fairness only concerns the honest party's utility, so an adversary who aborts without revealing his preference cannot hurt maximin fairness. Therefore, the strategy of first asking each party to reveal his preference, and then letting the non-aborting parties to execute a fair public preference protocol works directly for maximin fairness.

Theorem 17 (Possibility of perfect maximin fairness for 3 parties and fail-stop adversaries.). *There exists a 3-party private preference coin toss protocol that achieves perfect maximin fairness against any fail-stop adversaries.*

8 Related Work

Related works on strongly fair coin toss [19, 27] as well as Blum's notion of weak fair coin toss [16] have been discussed earlier in Section 1. In this section, we discuss additional related work.

Game theory and cryptography. Historically, game theory [36,48] and multi-party computation [27, 52, 53] were investigated by separate communities. Some recent efforts have investigated the marriage of game theory and cryptography (see the excellent surveys by Katz [37] and by Dodis and Rabin [24]). This line of work has focused on two broad types of questions:

- First, a line of works [1, 4–6, 32, 38, 49] investigated how to define game-theoretic notions of security (as opposed to cryptography-style security notions) for multi-party computation tasks such as secret sharing and secure function evaluation. Existing works consider a different notion of utility than us: specifically, these works make (a subset to all of) the following assumptions about players' utility: players prefer to compute the function correctly;

further, they prefer to learn secrets, and prefer that other players do not learn secrets. These works then investigate how to design protocols such that rational players will be incentivized to follow the honest protocol.
- Second, a line of work has asked how cryptography can help traditional game theory. Particularly, many classical works in game theory [36, 48] assumes the existence of a trusted mediator — and recent works have shown that under certain conditions, this trusted mediator can be implemented using cryptography [9, 23, 29, 34].

In this paper, we investigate game-theoretic notions of fairness for coin toss protocols. Our notions are novel in comparison with the aforementioned related work. First, to the best of our knowledge, we are the first to apply game theory to coin toss protocols, and asking whether we can circumvent known impossibilities [19] by considering rational players. Second, the fairness notions proposed in this paper are novel and to the best of our knowledge have not been investigated before for multiple parties. Specifically, we consider a natural notion of utility for coin toss protocols, where players have a preference over the outcome of the coin toss. We require that an honest execution produces an unbiased coin (unless all parties prefer the same bit); however if one or more coalition(s) deviate from the honest protocol, the coin toss outcome need not be unbiased (but we want that certain fairness properties must be preserved). All notions of fairness defined in the paper consider *corrupt majority* — since in the case of honest majority, strongly fair coin toss is known to be possible assuming standard cryptography assumptions [27], and the standard strong fairness notion implies all game-theoretic notions considered in this paper. In comparison, most earlier works [1,4–6,9,23,29,32,34,38,49] at the intersection of cryptography and game theory consider only the popular Nash equilibrium notion that is concerned about coalitions of size 1. Our fairness definitions are inspired by equilibrium notions in game theory that resist coalitions in various capacities [15, 35].

Other notions of fairness. Our work is inspired by the study of new, financially motivated fairness notions in blockchains and cryptocurrency applications [3, 8, 13, 22, 39–41, 45]. Several recent works [13, 22, 39, 40] show that to achieve a suitable notion of financial fairness, the protocol may require that parties place collateral on the blockchain to participate, and misbehaving parties can be penalized by taking away their collateral. Among these works, the most closely related to ours are those that investigate lottery-style protocols [8, 13, 22, 39, 45]. While earlier works [3, 13] require quadratic amount of collateral, more recent works [8, 45] showed that it is possible to realize fair lottery in the presence of a blockchain (i.e., a broadcast medium with identifiable abort) requiring no collateral at all, by relying on a folklore tournament-tree approach. Interestingly, although not explicitly noted, all these works on fair lottery over a blockchain [8, 13, 22, 39, 45] adopt a game theoretic notion of fairness, that is, although a deviating coalition can bias the outcome of toss of the n-sided dice, such bias must be towards a direction that harms the perpetrators. In fact, the implicit fairness notion in these papers is equivalent to our notion of maximin

fairness and cooperative-strategy-proof (CSP) fairness — for 0-sum games like a lottery, these two notions are equivalent.

Other relaxations of strong fairness have also been considered for coin toss and multi-party computation. For example, several works [2, 7, 10, 11, 14, 17, 19–21, 28, 30, 31, 46] consider a notion of ϵ-fairness, i.e., the adversary can bias the coin by at most a non-negligible ϵ amount. Moran et al. [46] showed that for general R, there is an R-round, 2-party coin toss protocol that satisfies $O(1/R)$-fairness — and this is optimal since Cleve [19] showed that for every R-round 2-party coin toss protocol, there exists an efficient adversary that can bias the honest party's outcome by at least $\Omega(1/R)$.

9 Conclusion

In this paper we proposed several natural, game theoretic notions of fairness for multi-party coin toss protocols. In the case of two parties, all of these notions equate to Blum's notion of weakly fair coin toss [16]; however, for more than 2 parties, these notions differ in strength and lead to different feasibility and infeasibility results. We summarize the strengths of various notions from strongest to weakest (for general n and mixed preference profiles).

Maximin $\neq$ **Cooperative-Strategy-Proof (CSP)** $>$ **Cooperative-Coalition-Proof (CCP)** $=$ **Strong Nash Equilibrium (SNE)** $>$ Coalition-Proof $>$ Nash

Among the above notions, we show broad feasibility results for SNE-fairness (which directly implies feasibility for coalition-proof equilibrium and Nash Equilibrium too). For other notions, we give a complete characterization of their feasibilities and infeasibilities — and for all of them we prove infeasibilities for amply divided preference profiles.

Note that among these notions, cooperative-strategy-proof and cooperative-coalition-proof are new notions first proposed in this paper — although we study them in the context of coin toss protocols, they would make sense for general games with transferrable utilities too. Finally, although maximin fairness is incomparable to CSP-fairness in general, the two are equivalent for balanced preference profiles (analogous to zero-sum games).

References

1. I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *PODC*, 2006.
2. B. Alon and E. Omri. Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In *TCC*, 2016.
3. M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek. Secure Multiparty Computations on Bitcoin. In *S&P*, 2013.

4. G. Asharov, R. Canetti, and C. Hazay. Towards a game theoretic view of secure computation. In *Eurocrypt*, 2011.
5. G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. In *CRYPTO*, 2009.
6. G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. *Journal of Cryptology*, 24(1), 2011.
7. B. Awerbuch, M. Blum, B. Chor, S. Goldwasser, and S. Micali. How to implement bracha's o (log n) byzantine agreement algorithm. *Unpublished manuscript*, 1985.
8. M. Bartoletti and R. Zunino. Constant-deposit multiparty lotteries on bitcoin. In *Financial Cryptography and Data Security*, 2017.
9. A. Beimel, A. Groce, J. Katz, and I. Orlov. Fair computation with rational players. `https://eprint.iacr.org/2011/396.pdf`, full version of Eurocrypt'12 proceeding version by Groce and Katz, 2011.
10. A. Beimel, I. Haitner, N. Makriyannis, and E. Omri. Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. Technical Report TR17-168, Electronic Colloquium on Computational Complexity, 2017. 7, 2017.
11. A. Beimel, E. Omri, and I. Orlov. Protocols for multiparty coin toss with a dishonest majority. *Journal of Cryptology*, 28(3):551–600, 2015.
12. M. Ben-or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC*, 1988.
13. I. Bentov and R. Kumaresan. How to Use Bitcoin to Design Fair Protocols. In *CRYPTO*, 2014.
14. I. Berman, I. Haitner, and A. Tentes. Coin flipping of any constant bias implies one-way functions. *JACM*, 65(3):14, 2018.
15. B. Bernheim, B. Peleg, and M. DWhinston. Coalition-proof nash equilibria i. concepts. *Journal of Economic Theory*, 42(1), 1987.
16. M. Blum. Coin flipping by telephone. In *CRYPTO*, 1981.
17. N. Buchbinder, I. Haitner, N. Levi, and E. Tsfadia. Fair coin flipping: Tighter analysis and the many-party case. In *SODA*, 2017.
18. K.-M. Chung, Y. Guo, W.-K. Lin, R. Pass, and E. Shi. Game theoretic notions of fairness in multi-party coin toss. Cryptology ePrint Archive, 2018.
19. R. Cleve. Limits on the security of coin flips when half the processors are faulty. In *STOC*, 1986.
20. D. Dachman-Soled, Y. Lindell, M. Mahmoody, and T. Malkin. On the black-box complexity of optimally-fair coin tossing. In *TCC*, 2011.
21. D. Dachman-Soled, M. Mahmoody, and T. Malkin. Can optimally-fair coin tossing be based on one-way functions? In *TCC*, 2014.
22. K. Delmolino, M. Arnett, A. E. Kosba, A. Miller, and E. Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *Financial Cryptography and Data Security*, 2016.
23. Y. Dodis, S. Halevi, and T. Rabin. A cryptographic solution to a game theoretic problem. In *CRYPTO*, 2000.
24. Y. Dodis and T. Rabin. Cryptography and game theory. In *Algorithmic Game Theory*, 2007.
25. B. Games. One-night werewolf.
26. J. A. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Eurocrypt*, 2015.
27. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *STOC*, 1987.

28. S. D. Gordon and J. Katz. Partial fairness in secure two-party computation. *J. Cryptology*, 25(1):14–40, 2012.
29. A. Groce and J. Katz. Fair computation with rational players. In *Eurocrypt*, 2012.
30. I. Haitner and E. Omri. Coin flipping with constant bias implies one-way functions. *SIAM Journal on Computing*, 43(2):389–409, 2014.
31. I. Haitner and E. Tsfadia. An almost-optimally fair three-party coin-flipping protocol. *SIAM Journal on Computing*, 46(2):479–542, 2017.
32. J. Halpern and V. Teague. Rational secret sharing and multiparty computation. In *STOC*, 2004.
33. R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *FOCS*, 1989.
34. S. Izmalkov, S. Micali, and M. Lepinski. Rational secure computation and ideal mechanism design. In *FOCS*, 2005.
35. R. J.Aumann. Acceptable points in general cooperative n-person games. Contributions to the Theory of Games IV", Princeton Univ. Press, Princeton, N.J., 1959.
36. R. J.Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1), 1974.
37. J. Katz. Bridging game theory and cryptography: Recent results and future directions. In *TCC*, 2008.
38. G. Kol and M. Naor. Cryptography and game theory: Designing protocols for exchanging information. In *TCC*, 2008.
39. A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *S&P*, 2016.
40. R. Kumaresan and I. Bentov. How to Use Bitcoin to Incentivize Correct Computations. In *CCS*, 2014.
41. R. Kumaresan and I. Bentov. Amortizing secure computation with penalties. In *CCS*, 2016.
42. H. Lin and R. Pass. Constant-round nonmalleable commitments from any one-way function. *J. ACM*, 62(1):5:1–5:30, 2015.
43. H. Lin, R. Pass, and M. Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *TCC*, 2008.
44. H. K. Maji, M. Prabhakaran, and A. Sahai. On the computational complexity of coin flipping. In *FOCS*, 2010.
45. A. Miller and I. Bentov. Zero-collateral lotteries in bitcoin and ethereum. In *EuroS&P Workshops*, 2017.
46. T. Moran, M. Naor, and G. Segev. An optimally fair coin toss. *J. Cryptol.*, 29(3):491–513, July 2016.
47. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
48. J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2), 1951.
49. S. J. Ong, D. C. Parkes, A. Rosen, and S. P. Vadhan. Fairness with an honest minority and a rational majority. In *TCC*, 2009.
50. R. Pass, L. Seeman, and A. Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Eurocrypt*, 2017.
51. R. Pass and E. Shi. Rethinking large-scale consensus. In *CSF*, 2017.
52. A. C.-C. Yao. Protocols for secure computations. In *FOCS*, 1982.
53. A. C.-C. Yao. How to generate and exchange secrets. In *FOCS*, 1986.