

# Rational Secret Sharing Under Fairness Frameworks

Jessica Shi, Evan Wildenhain  
COS 521, Fall 2017

## 1. Introduction

### 1.1. Background and motivation

Game theory is a classic and well-studied field that provides much of the framework for deciding rational actions based off of utilities and valuations. One important subtlety in analyzing game theory is that it is often beneficial to have a *trusted mediator*. In a typical strategic-form game, we often consider the *Nash equilibria* of such a game, where players choose moves independently of other players. Players can in fact receive higher utilities if they play according to a *correlated equilibria*, which is much more general than a Nash equilibrium and requires only that players would not wish to deviate from a given strategy given that other players do not deviate from their assigned strategies [4]. However, such equilibria require a trusted mediator to calculate the correct strategies and provide them to each player. The issue with depending on such a trusted mediator is that in reality, a honest third party may not necessarily exist.

This brings us to a classical result in cryptography. In cryptography, there is the construct of *multiparty computation*, where multiple parties each have secret inputs, and would like to calculate functions without revealing each of their inputs. This concept was first introduced by Yao [29], in particular in reference to the *millionaire problem*, where two millionaires would like to compute who is richer without revealing information regarding their wealth outside of that fact.

It follows linearly that we could potentially replace trusted mediators with a multiparty computation framework. Indeed, Dodis *et al.* [13] showed that if prior to a game we allowed a “cheap talk” phase where players could freely communicate<sup>1</sup>, then given a completely *fair* and secure multiparty computation scheme, replacing trusted mediators with such a scheme will translate any correlated equilibrium of the original game into a computational equilibrium of the new game.

One of the main issues in this model is *fairness*. A multiparty computation scheme is fair if either all players obtain their desired function results or no players do. Note that in cryptography, players are often considered to be either honest or dishonest; a well-known impossibility result states that in general, if a majority of parties are not honest, then complete fairness is impossible [11]. Note that some specific non-trivial functions can still be computed fairly without an honest majority [16], and the impossibility result only applies in general.

It is important, then, to consider what occurs during cheap talk if we do not have the guarantee of a fair scheme. Moreover, in a game theoretic model, it makes much more sense to consider players to be rational, rather than honest or dishonest. Dodis *et al.* [13] show a schematic for computing a correlated equilibrium between two parties, assuming a multiparty scheme that is secure but not fair and assuming that both parties act rationally. However, their constructs do not extend to more than two parties in general.

---

<sup>1</sup>This name arises because the phase costs and is worth nothing; indeed, players are free to lie in this phase.

Other results in this realm consider variations such as assuming parties cannot privately collude (so in the cheap talk phase, all messages are broadcast) [22, 20, 19] or investigating results of *partially fair* multiparty computation schemes [14, 23].

In this paper, we focus on formalizing the concept of rationality in the cheap talk phase. In particular, we consider the multiparty computation scheme to be a game in itself, and explore what it means for players to act rationally under certain equilibrium notions. Importantly, Shoham and Tennenholtz [28] introduce the concept of *NCC functions*, which are functions for which telling the truth is a Nash equilibrium. However, this heavily restricts the class of functions which we can use in the cheap talk phase, and moreover, Nash equilibrium may not be the correct equilibrium concept to consider; there may be refinements of Nash equilibrium that are more appropriate for this analysis.

In fact, Halpern and Teague [18] showed that one of the building blocks of many multiparty computation schemes,  $t$ -out-of- $n$  secret sharing, is impossible to accomplish under iterated deletion of weakly dominated strategies in the deterministic case. As a direct result of this, any deterministic multiparty computation scheme that requires shared-secret reconstruction is impossible under iterated deletion of weakly dominated strategies.

Some results related to this include Abraham *et al.* [1], who extend this problem to allow for coalitions and agents with unknown/nonstandard utilities, Lysyanskaya and Triandopoulos [24], who consider a *mixed-behavior* model where people are either rational or adversarial, and Gordon and Katz [17] who show that if in secret sharing we assume rational dealers (rather than honest dealers), then there exists a probabilistic  $t$ -out-of- $n$  secret sharing scheme.

## 1.2. Our results

In this paper, we focus on Halpern and Teague’s [18] results, albeit under a slightly different equilibrium concept. We consider iterated deletion of weakly dominated strategies in every subgame of deterministic  $t$ -out-of- $n$  secret sharing schemes (meaning, we restrict the ordering of iterated deletion to the natural order in subgame hierarchies). Their proof in fact shows the same impossibility result under this equilibrium concept, and we will reiterate the relevant portions of their proof.

Moreover, in cryptographic literature, there are multiparty computation frameworks that attempt to introduce fairness by using utilities outside of classical cryptography. We focus on two such mechanisms, namely *gradual release* [15] and a compensation framework based on Bitcoin/Ethereum [21], which introduce penalties for failure to contribute honestly through resource and money incentives respectively. Note that both of these mechanisms allow for fair multiparty computation even if there is no honest majority. We show that gradual release, under a game theoretic analysis with rational players and with utilities adjusted to take into account resource incentives, is impossible under iterated deletion of weakly dominated strategies in every subgame. However, the compensation framework under a symmetric analysis results in a fair secret sharing scheme in which all players are incentivized to participate.

We further note that the negative result for gradual release case implies an impossibility result for multiparty computation schemes dependent on secret sharing, parallel to the analysis by Halpern and Teague [18]. The positive result for the compensation framework implies that it is possible for such multiparty computation schemes to be rationally carried out, although does not necessarily imply definitively positive results in multiparty computation. There are also further issues in the compensation framework, in terms of correctly pricing cost incentives such that the utilities we

assume hold.

### 1.3. Outline

In Section 2, we introduce some basic game theoretic and cryptographic notions. We also discuss the implications of results in rational secret sharing on multiparty computation schemes. In Section 3, we set up Halpern and Teague’s [18] framework for analyzing secret sharing, and we prove the impossibility of deterministic secret sharing schemes. In Section 4, we briefly introduce the notion of gradual release [15] and we prove the impossibility of deterministic secret sharing schemes under a gradual release framework. Finally, in Section 5, we briefly introduce the notion of a compensation framework using Bitcoin/Ethereum [21] and we prove the success of deterministic secret sharing schemes under a compensation framework, under iterated deletion of weakly dominated strategies in every subgame. We also discuss the restrictions of this result, specifically problems in pricing cost incentives.

## 2. Preliminaries and definitions

### 2.1. Game theory

We begin by introducing some standard game theoretic notation, with minor adaptations for simplicity. The games we consider here are *dynamic*, that is to say, they consist of multiple stages, and we represent them in *extensive form*.

An *extensive form* game consists of [26]:

1. *players*:  $N = [n]$
2. *histories*:  $H$ , where  $h \in H$  is a sequence of moves up to a given point in time and  $\emptyset \in H$  represents the start of the game
3. *available actions*:  $A(h)$  for all  $h \in H$ 
  - (a) *terminal histories*:  $Z \subset H$  are such that  $A(z) = \emptyset$  for all  $z \in Z$
4. *player assignment function*:  $P(h) \in N$ , which determines which player moves at  $h \in H \setminus Z$
5. *information sets*:  $I$ , which are given as a partition of  $H$  such that for  $h, h' \in X \in I$ ,  $P(h) = P(h')$  and  $A(h) = A(h')$ ; in other words, player  $P(h)$  cannot distinguish between  $h$  and  $h'$
6. *payoffs*:  $\{U_i(z)\}_{i \in N}$  at every terminal history  $z \in Z$

This game can be thought of in the form of a *game tree*, where each node is a history and each edge represents the available actions at that history.

We divide a secret sharing or multiparty computation scheme into *rounds*, where at each round, each player gives information to some subset of other players (a player may choose to give no information in a round). Note that information exchange occurs synchronously within each round.

For simplicity, we collapse our game tree so that each level of the tree corresponds to a round of a game, and the edges out of each node represent the collection of actions performed in the corresponding round. Note that multiple nodes within a level may be within the same information set for a player  $i$ , but nodes across different levels are necessarily within different information sets for that player.

Now, a *strategy* for player  $i$  is a function that maps their information sets to actions (note that these actions may be taken probabilistically). We denote a strategy for player  $i$  by  $\sigma_i$ , and a strategy

over all players by  $\bar{\sigma} := (\sigma_1, \dots, \sigma_n)$ . We let  $\sigma_{-i}$  denote the tuple of all strategies  $\sigma_j$  for  $j \neq i$ , and abusing notation slightly, we define  $\bar{\sigma} = (\sigma_{-i}, \sigma_i)$ .

Each player has a *preference function* for strategies, which we denote by  $U_i(\bar{\sigma})$  for player  $i$ . Note that actions that players take can be represented as a path from the root node of the game tree to a leaf, so we can alternatively represent a strategy by such a path, which we call a *run* and denote by  $r$ , and  $U_i(r)$  is defined accordingly.

For the purposes of this paper, we consider primarily a refinement of Nash equilibrium, namely *iterated deletion of weakly dominated strategies* in every subgame. While Nash equilibrium is a well-known and traditional concept for considering rationality, in the context of secret sharing and multiparty computation, there are multiple strategies that lead to Nash equilibrium. The concepts behind iterated deletion of weakly dominated strategies mirror classic arguments for fairness in secret sharing, and as such we focus on this as a definition for rationality.

More formally, a player  $i$ 's strategy  $\sigma_i$  *weakly dominates*  $\sigma'_i$  if  $U_i(\sigma_{-i}, \sigma_i) \geq U_i(\sigma_{-i}, \sigma'_i)$  for all  $\sigma_{-i}$  and there exists  $\sigma'_{-i}$  such that  $U_i(\sigma'_{-i}, \sigma_i) > U_i(\sigma'_{-i}, \sigma'_i)$ . Intuitively, we should want honest participation in our protocols to be a Nash equilibrium in which no player is playing a weakly dominated strategy, as any such player would have no reason not to play a dishonest strategy instead.

In iterated deletion of weakly dominated strategies in extensive form games in every subgame, we make use of *backward induction*, where we start at the end of the game (at the leaves of the game tree), and at each information set, we delete all strategies that are weakly dominated. We proceed with this process level-by-level up the game tree, until we reach the root. The strategies which remain are our equilibrium strategies. Note that it is important that we delete all weakly dominated strategies at each step, since otherwise, the order in which we delete weakly dominated strategies may affect the outcome of this iterated deletion process [26]. However, we are explicitly ordering our iterated deletion by the natural subgame hierarchy within our extensive form game (note that every level, by construction, represents a subgame).

## 2.2. Secret sharing and multiparty computation

**2.2.1. Secret sharing** We now discuss the basic setup for *t-out-of-n secret sharing*. A dealer holds a secret message  $m$ , which he splits into shares  $(m_1, \dots, m_n)$  such that anyone with  $t$  shares can reconstruct the message, but anyone with  $(t - 1)$  shares cannot. Moreover, anyone with  $(t - 1)$  shares “learns nothing” about  $m$ ; more concretely, given two messages  $m \neq m'$ , for all  $S \subseteq [n]$  such that  $|S| < t$ , the distribution of  $\{m_i\}_{i \in S}$  must be computationally indistinguishable from the distribution of  $\{m'_i\}_{i \in S}$ . Note that in this scenario, we commonly think of  $n$  people, where the dealer gives person  $i$  share  $m_i$ .

Note that in cryptography, we primarily consider secret sharing among  $n$  honest parties, less than  $t$  of which have been compromised by an outside adversary. In this sense, the message  $m$  is safe from outside adversaries and the honest parties can meet to jointly reconstruct the message; all honest parties follow protocols completely, and as such reconstruction is straightforward in that they will necessarily reveal their shares to other honest parties.

In this sense, reconstruction is not necessarily *fair*. A scheme is fair if either all players obtain the secret message or no players do. However, in this scenario, clearly  $(t - 1)$  players can reveal their shares and the final player can abstain from revealing his share, and leave with the secret message. This is a concept that is considered more heavily in multiparty computation, and we will discuss this further in Section 2.2.2. Note that this is a particularly relevant point in the context of

rationality, where we have  $n$  parties who may or may not agree to send each other shares to allow for reconstruction, depending on whether it is rational for them to do so. If a party gains at least  $(t - 1)$  other shares, then it will be able to reconstruct the message, although that party may not necessarily reveal its shares to other parties. This is the basic scenario that we investigate in Sections 3.1, 4.2, and 5.3.

As an example, one well-known secret sharing scheme is Shamir’s secret sharing [27], where the dealer chooses a random degree  $(t - 1)$  polynomial  $f$  such that  $f(0) = m$  and where shares are given by  $m_i := f(i)$ . Other notable schemes have been introduced by Blakley [7], Asmuth and Bloom [3], and Mignotte [25].

**2.2.2. Multiparty computation** Secret sharing forms the basis of many protocols for *multiparty computation*; in this paper, we focus specifically on schemes that depend on secret sharing [8, 10, 12, 14, 23]. In multiparty computation, we have  $n$  parties, each with a private input  $x_i$ , who would like to jointly compute a function  $f(x_1, \dots, x_n)$  without revealing their individual inputs (excepting information revealed in the function output).

As a concrete example, the *BGW* protocol is a general multiparty computation scheme in which each player acts as a dealer and distributes shares of their private input to all of the other players. These shares are computed such that each player can use them to compute a share of the final output [6, 2]. Then,  $t$  players can combine their shares to receive the final output. This protocol encapsulates a few important notions, which we will briefly discuss here.

First, it is important to note that in the multiparty computation schemes that we consider, parties are expected to be “honest-but-curious.” That is to say, we assume that all parties engage in the protocol correctly, and do not attempt to sabotage the protocol by sending false information; however, players may attempt to compute as much as possible with the information that they receive. In particular, protocols based on  $t$ -out-of- $n$  secret sharing are secure against a group of  $< t$  *passive adversaries* (also known as *semi-honest security*); such adversaries follow the protocol, but may attempt to pool their shares and collude to gain any outside information, and in particular, to gain information about the shares of other players.

Note that this assumption is a direct extension of the assumptions in secret sharing, where we assume that the dealer provides honest, authenticated shares so that players cannot send each other falsified shares.<sup>2</sup> As such, when we consider rational players in multiparty computation, we assume that such players are “rational-but-not-malicious”; in this sense, players can decide to either send honest information or abstain from participating, but cannot send false information.

Note that our schemes do afford us privacy and security against outside adversaries. Privacy is the notion that any given player learns nothing more about the other players’ inputs than the function output; there are ways to formalize this, specifically in *simulation security*, but this is outside the scope of our paper. Security against outside adversaries occurs in the sense that for a multiparty computation scheme based on  $t$ -out-of- $n$  secret sharing, if an adversary gains the views of  $< t$  players, then that adversary “learns nothing” about the final output.

Finally, we are interested in the concept of fairness, as mentioned in Section 2.2. Importantly, in classical cryptography where players are either fully honest or not, complete fairness is impossible in general if a majority of the parties are not honest (this is due to a well-known result by Cleve [11]); this is one of the motivations for considering rational parties instead. However, in Section 3.1

---

<sup>2</sup>There are extensions to secret sharing that address the possibility of malicious players, as in *verifiable secret sharing*. This is outside the scope of this paper.

(and in Halpern and Teague’s [18] result), one of the core reasons for the failure of rational secret sharing is due to an inherent lack of fairness; as such, we consider schemes outside of classical notions, which incorporate fairness even with no honest majority by allowing other motivations, such as resources as in Section 4.2 and money as in Section 5.3.

Note that Cleve’s [11] result does not preclude specific non-trivial functions from being computed fairly without an honest majority; Gordon *et al.* [16] demonstrated fair protocols for certain non-trivial functions in a two-party computation setting.

**2.2.3. Implications of secret sharing on multiparty computation** We digress here to make a point about how results in rational secret sharing generalize to multiparty computation. There are well-known schemes for multiparty computation that require fair secret sharing (and are deterministic and have an upper bound of running time<sup>3</sup>) [8, 10, 12, 14, 23], and any secret sharing scheme that is proven to be impossible under rationality (as in Halpern and Teague’s results [18]) will likewise render those multiparty computation schemes impossible under rationality.

On the other hand, if we have a secret sharing scheme in which players will rationally carry out the protocol, then this protocol can be applied to such multiparty schemes dependent on fair secret sharing. While this does not necessarily prove that these multiparty computation schemes will be rationally carried out, and indeed in the BGW protocol this does not solve the issues of requiring a “honest majority”<sup>4</sup>, this is a step towards eventually achieving such results.

Here, we briefly formalize the point that impossibility in rational secret sharing implies impossibility in rational multiparty computation. In multiparty computation, players exchange shares of their own input and perhaps other information to eventually compute a function. Let  $I_1, \dots, I_N$  represent each atomic piece of information that are of interest to a player in performing the multiparty computation.

The utilities for receiving certain pieces of information here are somewhat more complicated to generalize for cases where players have other values such as time and money, as in Sections 4.2 and 5.3; however, they follow as a direct extension of the utilities given in those sections. We will focus here on the classic case, given by Halpern and Teague [18] and as an extension of the utilities for secret sharing in Section 3.1. In this setting, each player cares first that they learn the secret, and second that as few other people as possible learn the secret (roughly); there are no other factors affecting each player’s utility.

More formally, given a run  $r$  on the game tree, we define  $\text{info}(r) := (\mathcal{I}_1, \dots, \mathcal{I}_n)$ , where  $\mathcal{I}_i \subseteq \{I_1, \dots, I_N\}$  represents the information that player  $i$  receives in  $r$ . We let  $\text{info}_i(r) := \mathcal{I}_i$ . The relevant utilities are given as follows:

- U-1.  $U_i(r) = U_i(r')$  if  $\text{info}(r) = \text{info}(r')$
- U-2.  $U_i(r) \geq U_i(r')$  if  $\text{info}_i(r) \supseteq \text{info}_i(r')$  and  $\text{info}_j(r) \subseteq \text{info}_j(r')$  for  $j \neq i$
- U-3.  $U_i(r) > U_i(r')$  if  $\text{info}_j(r) \subset \text{info}_j(r')$  for some  $j \neq i$ ,  $\text{info}_{j'}(r) = \text{info}_{j'}(r')$  for all  $j' \neq j$ , and  $U_j(r) < U_j(r')$
- U-4.  $U_i(r'_1) < U_i(r'_2)$  if  $\text{info}_i(r_1) = \text{info}_i(r'_1)$ ,  $\text{info}_i(r_2) = \text{info}_i(r'_2)$ ,  $\text{info}_j(r_1) = \text{info}_j(r_2)$  for all  $j \neq i$ ,  $\text{info}_j(r'_1) = \text{info}_j(r'_2)$  for all  $j \neq i$ , and  $U_i(r_1) < U_i(r_2)$

<sup>3</sup>These are conditions that we require in our results in Sections 3.1, 4.2, and 5.3.

<sup>4</sup>In the BGW protocol, even with rational secret sharing, we still require that  $t < n/2$ , which is a condition that arises from the Lagrange interpolation that is used within the protocol.

These utilities represent generalizations of the utilities for classic secret sharing in Section 3.1, and it can be checked that the utilities necessary for classic secret sharing hold as long as these utilities hold.

In this sense, if it is impossible to carry out a secret sharing protocol under a certain framework, then it is impossible to complete the secret sharing necessary for such multiparty computation schemes.

### 3. Classic setting

In this section, we consider a classic cryptographic setting, where players' utilities are solely defined by who receives knowledge of the secret message. The results here follow from Halpern and Teague's [18] work, although we introduce only a portion of their proof.

#### 3.1. Secret sharing

In this subsection, we consider a  $t$ -out-of- $n$  secret sharing scheme, and prove that as long as there exists a known bound on running time, there is no deterministic mechanism for such secret sharing under rationality. This result was first proven by Halpern and Teague [18], using backwards induction and a full classification of types of information exchange. We instead use a slightly different concept of rationality and show an impossibility result using only backwards induction.

**3.1.1. Setup** We begin with assumptions similar to those of Halpern and Teague [18]. We assume that a dealer issues atomic and authenticated shares, so that players cannot subdivide or lie about their shares.<sup>5</sup> We will begin with a model in which there is a single secret message,  $m$ , and each player  $i$  has share  $m_i$  of that message. Note that  $t$  shares are sufficient to reconstruct  $m$ .

At each node in our game tree, a player  $i$  can

- Give a subset of other players their authenticated share,  $m_i$
- Give a subset of other players an authenticated share that they have received from another player  $j \neq i$ ,  $m_j$
- Give a subset of other players an authenticated share that they have received from another player  $j \neq i$ ,  $m_j$ , that is also signed by a different player  $k \neq i, j$

The main restriction is that when sending shares,  $i$  must send valid authenticated shares. In this sense, when a person  $k$  receives a share, they can be certain that the shares are true. Note that  $i$  may also send information about who knows which shares, since clearly if person  $k$  signed a share  $m_j$ , then person  $k$  knows that share. However, this information must be accompanied by the share in question itself.

The utilities of a run through the game tree are given solely by which players can compute the secret message  $m$ , and these are taken directly from Halpern and Teague's [18] setup. Given a run  $r$ , we define  $\text{info}(r) := (s_1, \dots, s_n)$  where  $s_i = 1$  if person  $i$  learns  $m$  and  $s_i = 0$  otherwise. We let  $\text{info}_i(r) := s_i$ . The utilities are given as follows:

$$\text{U-1. } U_i(r) = U_i(r') \text{ if } \text{info}(r) = \text{info}(r')$$

$$\text{U-2. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) > \text{info}_i(r')$$

---

<sup>5</sup>Note that earlier, in Section 2.2.2, we have seen that this assumption does not necessarily hold in multiparty computation. However, since we are proving an impossibility result, it is alright for us to make this assumption.

U-3.  $U_i(r) > U_i(r')$  if  $\text{info}_i(r) = \text{info}_i(r')$ ,  $\text{info}_j(r) \leq \text{info}_j(r')$  for all  $j \neq i$ , and there exists some  $j$  such that  $\text{info}_j(r) < \text{info}_j(r')$

In this sense, each person  $i$  values first that they learn the secret, and second that as few other people as possible learn the secret (roughly). This gives us all of the necessary information to proceed to our impossibility result.

### 3.1.2. Analysis

**Theorem 3.1.** *If the players' utilities satisfy U-1 – U-3, then there is no deterministic mechanism for  $t$ -out-of- $n$  secret sharing in which the game tree has a commonly known bound and using iterated deletion of weakly dominated strategies in every subgame, some player learns the secret message.*

*Proof.* Suppose that our game tree has  $\ell$  levels, where  $\ell$  is given by the commonly known bound on running time. We will use backwards induction on the level, and in particular, we will start with the inductive step. The base case will be clear by a similar argument. At each stage in our induction, we will delete all weakly dominated strategies at the corresponding level.

The focus of our analysis will be on information sets containing the strategy in which no one sends any shares to anyone. Call the strategy of doing nothing  $s$ , and let the path through the tree corresponding to  $s$  be denoted by the nodes  $(n_0, \dots, n_\ell)$ , where  $n_0$  is the root and  $n_\ell$  is a leaf node. Denote the information set for player  $i$  in level  $m$  containing node  $n_m$  by  $I_{i,m}$ .

Assume that at level  $m$ , for every player  $i$  in every information set, doing nothing is never weakly dominated, and in particular, the utilities given by iterated deletion of weakly dominated strategies up to level  $m$  are precisely the utilities of doing nothing up to level  $m$ .<sup>6</sup> Moreover, assume that for each  $i$ , in  $I_{i,m}$ , the sole weakly dominant strategy is for  $i$  to do nothing. We now show that these two properties hold in level  $m - 1$ .

Fixing person  $i$ , it is easy to see that  $i$  can never have a lower utility by sharing no information, and as such, sharing no information will never be weakly dominated. More precisely, because of our iterated deletion through level  $m$ , no matter what  $i$  sends or does not send,  $i$ 's utilities will be as if no exchanges happen after this round. Since sending information can only (potentially) decrease  $i$ 's utility, it is clear that sending nothing is never weakly dominated, and in fact, the utility given by deleting weakly dominated strategies will be precisely the utility given in which  $i$  does nothing.

It remains to show that in the information set  $I_{i,m-1}$ , the sole weakly dominant strategy is for  $i$  to do nothing. Note that  $i$  has no information other than  $m_i$ , since otherwise  $i$  would not be in this information set. If  $i$  chooses to send  $m_i$  to any person  $j$ , then at node  $n_{m-1}$ , there is an edge corresponding to the case in which  $t - 2$  other people choose to send their shares to  $j$  ( $\delta_{-i}$ ), and by our inductive hypothesis, the utility of choosing that edge is as if no other exchanges occur following this (since all other strategies have been deleted). It is clear, then, that for person  $i$ , it is strictly better to send no shares ( $\delta'_i$ ) than to send any shares ( $\delta_i$ ) at node  $n_{m-1}$ , that is to say,

$$U_i(\delta_{-i}, \delta'_i | n_{m-1}) < U_i(\delta_{-i}, \delta_i | n_{m-1}).$$

This trivially extends if  $i$  chooses to send any number of messages to anyone. Moreover, as explained before, at any other node in the information set, whether  $i$  sends shares or not cannot have a negative

---

<sup>6</sup>In this sense, it does not matter which equilibrium strategy we choose to collapse that information set to at any given level; it suffices to use the utilities given by doing nothing.

impact on  $i$ 's utility, since  $i$ 's utility only decreases with sharing more information. Thus, for all  $p \in I_{i,m-1}$  and for all  $\sigma_{-i}$ , we have

$$U_i(\sigma_{-i}, \delta_i | p) \leq U_i(\sigma_{-i}, \delta'_i | p).$$

As such, doing nothing is the sole weakly dominant strategy for  $i$  within  $I_{i,m-1}$  at level  $(m-1)$ . This argument applies symmetrically for all people and their relevant information sets, and this concludes our inductive step. Note that the base case is argued in precisely the same way.

At node  $n_0$ , we note that since no one has any prior information, there is merely one information set left. This information set is necessarily  $I_{i,0}$  for all  $i$ , and as a result,  $s$  is the sole strategy that survives iterated deletion of weakly dominated strategies at every subgame. This concludes our proof.  $\square$

## 4. Gradual release setting

In this section, we consider a modified setting where players' utilities also depend on the resources and time necessary to obtain the secret message. We extend the impossibility results by Halpern and Teague [18], and demonstrate that there is no deterministic method for rational  $t$ -out-of- $n$  secret sharing in this framework (given a commonly known upper bound).

### 4.1. Gradual release

We focus on a mechanism known as *gradual release*, whereby complete fairness can be achieved even without an honesty majority. Colloquially, gradual release involves releasing secrets piece by piece, such that if a party aborts at any stage, the remaining parties can compute the secret in approximately the time it takes the aborting party to compute the secret. There are various initial results in this realm, such as from [8, 14, 5, 12].

We present here results by Garay *et al.* [15], who use gradual release formally to achieve fairness in multiparty computation. In particular, Garay *et al.* note that the main fairness issue with multiparty computation schemes lies in a *revelation* phase, where parties each reveal their secret shares to construct the final output. Their gradual release scheme addresses this phase.

**4.1.1. Commit-prove-fair-open** The basic structure for their scheme involves the “*commit-prove-fair-open*” scheme,  $\mathcal{F}_{\text{CPFO}}$ , which consists of three phases. In the commit phase, every party  $i$  broadcasts a commitment to a value  $x_i$ . In the prove phase, every party broadcasts a proof  $y_i$  such that  $R(x_i, y_i) = 1$  for some relation  $R$ . Finally, in the open phase, each party opens  $x_1, \dots, x_n$  simultaneously. Note that the opening phase may take multiple rounds, but this simultaneous opening guarantees fairness.

**4.1.2. Timelines** To implement this scheme, we introduce the notion of a *timeline*.

Let  $N = pq$  be a Blum integer, that is to say,  $p, q$  are distinct primes congruent to  $3 \pmod{4}$ . Define  $G := (g, g^2, g^{2^2}, g^{2^3}, \dots, g^{2^k})$  where operations are taken in  $\mathbb{Z}_N$  and  $g$  is randomly chosen from  $\mathbb{Z}_N^*$ . We also define  $G[i] = g^{2^i}$ , and we assume that squaring takes one unit of time. Given a factorization of  $N$ , it is possible to compute  $G[i]$  for any  $i$  in polynomial time. However, it is postulated that without such a factorization, each squaring must be computed sequentially, and more strongly, that given  $a_1, \dots, a_{\ell+1}$  where  $|a_{\ell+1} - a_i| \geq 2^\ell$  for  $i \in [\ell]$  and given  $(G[a_1], \dots, G[a_\ell])$ ,  $G[a_{\ell+1}]$  appears pseudorandom [15]. This is known as the *yet-more-general BBS assumption* (YMG-BBS), and is in fact of the *generalized BBS assumption*, which was first introduced by Boneh and Naor [10].

We define a *decreasing timeline* to be  $T = \langle N, g, \vec{u} \rangle$ , where  $N$  is a Blum integer,  $g = G[0]$ , and  $u[i] = G[2^k - 2^{k-i}]$  for  $i \in [k]$ . In essence,  $u[k]$  appears pseudorandom given  $g$ , and takes exponential time to obtain given  $g$ .

Given a *master timeline*  $T$ , we define a *derived timeline* to be a timeline  $T' = \langle N, h, \vec{v} \rangle$  such that  $h = g^\alpha$  and  $v[i] = (u[i])^\alpha$  for  $\alpha \in \mathbb{Z}_{[1, (N-1)/2]}$ . We call  $\alpha$  the *shifting factor*. Importantly, assuming the composite decisional Diffie-Hellman (CDDH) assumption [9] and the YMG-BBS assumption,  $v[k]$  appears pseudorandom even given the entire master timeline [15].

**4.1.3. Implementing gradual release** Now, we combine the notion of a timeline with a commit-prove-fair-open scheme. We take a master timeline  $T$  to be a *common reference string* (CRS), known to all players.

In the commit phase, each player  $i$  derives a timeline  $T_i = \langle N, g_i, \vec{u}_i \rangle$  and broadcasts a timeline-commitment  $(g_i, x_i \cdot u_i[k])$ . Given  $x_i \cdot u_i[k]$ , any player  $j$  can *force-open* the commitment by repeatedly squaring  $g_i$ ; however, since  $u_i[k]$  appears pseudorandom, this is considered too costly to be possible.

In the prove phase, each player  $i$  gives a zero knowledge proof that they know the shifting factor for their derived timeline.

We have  $k$  rounds in the open phase. On round  $\ell$ , each player  $i$  broadcasts  $u_i[\ell]$  along with a zero knowledge proof that this is a valid point. If in any round a player aborts or fails to broadcast, then all players abort and force-open the timeline-commitments that they have received if it is feasible to do so using repeated squaring. Otherwise, if the distance to the end of the timeline is too large, all players simply do nothing. Note that in this case, the players who aborted the round prior will also be unable to reach the end of the timeline, since the information that they have received will still appear pseudorandom with respect to  $u_i[k]$  for all  $i$ . Garay *et al.* [15] proved that this scheme is fair.

Using this commit-prove-fair-open scheme in place of the revelation phase of multiparty computation schemes, we obtain fair multiparty computation. Details regarding security and the specific models in which a commit-prove-fair-open scheme may be used are outside the scope of this paper, but are given in [15].

## 4.2. Secret sharing

We now consider the concept of rational  $t$ -out-of- $n$  secret sharing under a gradual release setting. We prove that there is no deterministic method for secret sharing in this framework under iterated deletion of weakly dominated strategies in every subgame, provided that there exists a commonly known bound on running time.

**4.2.1. Setup** As in Section 3.1, we assume that shares are atomic and authenticated. Moreover, we enforce the gradual release protocol to some degree; while theoretically players can send timeline-commitments and proofs as they please, this results in a simple extension of the arguments made in Section 3.1, with some added utilities regarding the time it takes to recover the secret. This also poses issues with the protocol itself in terms of varying the number of rounds, since the essence of the protocol depends on the number of rounds it takes to complete the protocol versus the time it takes to force-open a timeline-commitment.

In this spirit, we focus on a model in which players either follow the gradual release protocol or abort. In particular, we have  $k + 1$  rounds in our game, where the first round represents the commit and prove phases of gradual release and the remaining rounds represent the open phase of gradual release. In each round, each player chooses other players to send its corresponding timeline-commitment and proof to. A player may alternatively choose to abort and force-open its

information. In this sense, we disallow communication of all other types of information.<sup>7</sup>

We assume that if a player  $i$  chooses not to send player  $j$  its timeline-commitment and proof in a given round, then player  $i$  cannot send player  $j$  timeline-commitments and proofs in any future rounds. As such, if a player receives  $< t$  timeline-commitments in a round, then we assume that the player force-opens its timeline-commitments from the previous round and aborts game participation.<sup>8</sup>

The utilities for our players are given in a similar way as in Section 3.1. There is, however, an added notion of time needed to obtain the secret. Given a run  $r$  in the game tree, let  $\text{time}(r)$  be a tuple  $(t_0, \dots, t_n)$ , where  $t_i$  denotes the time it takes player  $i$  to obtain the secret. More precisely, we define playing a round to take time 1 and squaring once to take time 1. Notably,

- If all players follow the protocol, then  $t_i = k + 1$  for all  $i$ .
- If player  $i$  force-opens,  $t_i$  is given by the sum of the number of rounds played and the number of squarings required to force-open its timeline-commitments.
- If player  $i$  receives  $< t$  commitments in the initial commit-prove round, then  $t_i = \infty$ .

Now, let  $\text{info}(r)$  be a tuple  $(s_1, \dots, s_n)$ , where  $s_i = t_i^{-1}$  (where we take  $\infty^{-1}$  to be 0). We have the following assumptions:

$$\text{U-1. } U_i(r) = U_i(r') \text{ if } \text{info}(r) = \text{info}(r')$$

$$\text{U-2. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) > \text{info}_i(r')$$

$$\text{U-3. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) = \text{info}_i(r'), \text{info}_j(r) \leq \text{info}_j(r') \text{ for all } j \neq i, \text{ and there is some } j \text{ such that } \text{info}_j(r) < \text{info}_j(r')$$

In this manner, each person values first that they learn the secret, second that they learn the secret quickly, and third that as few other people as possible learn the secret (roughly) with a preference for other people learning the secret as slowly as possible. We now proceed to our impossibility result.

#### 4.2.2. Analysis

**Theorem 4.1.** *If the players utilities satisfy U-1 – U-3, then there is no deterministic mechanism for  $t$ -out-of- $n$  secret sharing using a gradual release protocol in which the game tree has a commonly known bound and using iterated deletion of weakly dominated strategies in every subgame, some player learns the secret message.*

<sup>7</sup>It is important to note that it is somewhat difficult to formalize allowable outside communication while remaining within the gradual release protocol. The time it takes for communication between rounds of the protocol must in some way be limited by the time it takes to force-open timeline-commitments at that round, since otherwise it would defeat the purpose of gradual release. As we proceed deeper into the open phase, the time to force-open becomes polynomial, and the restrictions on outside discussion must reduce to reflect this change. We defer a more thorough analysis of this point.

<sup>8</sup>Technically, players can skip rounds and simply send timeline-commitments and proofs corresponding to later rounds. This, however, is against the spirit of gradual release, in the sense that players can ignore gradual release entirely and simply participate only in the final round. We require that players commit in previous rounds if they wish to continue the gradual release protocol.

*Proof.* The argument follows almost exactly from that in Theorem 3.1, with backwards induction; this is due to the nature of the utilities, and the fact that each person values that others learn the secret slowly.

In particular, at any given round, sharing no information will never be weakly dominated, since information shared only serves to assist other players in learning the message faster.

Moreover, in the information set containing the strategy  $s$  where no one does anything, not sharing information at any given round will always be strictly better than sharing information, in the case where  $t - 2$  other people also share their time-commitments and proofs (with the inductive hypothesis).

These two notions roughly encapsulate the basic premise of the backwards induction; we leave out a full proof here.  $\square$

## 5. Compensation setting

We now consider a viable alternative approach that does survive iterated deletion of weakly dominated strategies in every subgame; namely, we consider a scheme that constructs a payment-commitment mechanism in which parties that behave dishonestly (i.e. renege on their commitment to participation in the MPC protocol) compensate parties that participate honestly. As demonstrated by Kiayias *et al.* [21], it is possible to construct a "robust MPC protocol with compensation," which takes a semi-honest MPC protocol and augments it with a payment-commitment mechanism that guarantees that, within a constant number of coin-transfer and communication rounds, honest participants will either receive the desired output or receive a net coin profit. This mechanism can be implemented in practice using Ethereum.

Our own work is to show the extent to which this protocol-with-compensation scheme extends to the rational-participant setting, as Kiayias *et al.* merely show that honest parties receive compensation without considering the utilities of any of the participants. We demonstrate that this protocol is dominant-strategy-honest with sufficiently large coin values, but making the protocol tremble-resistant is only achievable for certain sets of utility functions for the participants.

### 5.1. The commitment ledger

The protocol described in [21] requires access to a *ledger* that supports special transactions that include conditions under which the transferred coins can be spent. Specifically, transactions placed in this ledger include a three-part validation predicate for deciding if the transaction is valid.

Let a transaction be defined as transferring coins from participant  $i$  to participant  $j$ . Then, the validation predicate is specified as follows:

1. *Time restriction:* An interval of time  $(\tau_-, \tau_+) \in \mathbb{Z} \times (\mathbb{Z}^+ \cup \{\infty\})$  where before time  $\tau_-$ , no party can spend the coins in the transaction; between time  $\tau_-$  and  $\tau_+$ , participant  $j$  can spend them (provided that parts 2 and 3 of the validation predicate are met), and after time  $\tau_+$ , the coins revert to being unrestrictedly owned by party  $i$ .
2. *Spending link:* An identifier  $\alpha$  for linked transactions, where a transaction from  $i$  to  $j$  transferring  $v$  coins is valid only if participant  $i$  has received a net gain of at least  $v$  coins in transfers with spending link ID  $\alpha$ . Kiayias *et al.* [21] use this to link together with transactions so as to facilitate reclaiming coins from transactions if a would-be recipient has violated the protocol.

3. *State-dependent condition*: A function  $\mathcal{R}$  from the current ledger-state, ledger-buffer, and transaction to be validated to  $\{0, 1\}$ , where only transactions that cause  $\mathcal{R}$  to evaluate to 1 are valid. This part is used to specify that participants in the protocol must participate correctly in the MPC or secret-sharing protocol in order to claim the funds committed to them.

## 5.2. The compensation protocol

The compensation protocol in [21] is designed to be composed with an arbitrary semi-honest protocol  $\pi_{SH}$ , where the only assumption required (beyond that the protocol is semi-honest secure) is that the protocol provide for verifying that participants have given correct input via zero-knowledge proofs. As described in Section 2.2.2, in our "rational-but-not-malicious" setting, players are assumed to be limited to either sending honest information or abstaining from participating; in this sense, we satisfy the semi-honest condition. Note that the semi-honest protocol is also assumed to terminate in a number of rounds with a known upper bound  $\ell$ .

**5.2.1. Setup** At time  $\tau = 0$ , the participants check that they have sufficient funds for the protocol, where "sufficient funds" means  $(n - 1) \cdot c$  coins, where  $c$  is the amount transferred in each individual transaction from a player  $i$  to a player  $j$ .

At time  $\tau = 1$ , every player  $i$  submits the following "commitment" transactions to the ledger: For each protocol round from  $r = 1, \dots, \ell$ , and each player  $j \neq i$ , there will be a transaction of  $c$  coins from  $i$  to  $j$  such that  $j$  receives the coins only if she claims them in round  $r$ .<sup>9</sup> These commitment transactions specify that in order for player  $j$  to claim their coins in round  $r$ , player  $j$  must place a "claiming" transaction in round  $r$  where its `aux` field contains  $j$ 's valid message for round  $r$  of  $\pi_{SH}$ , and the protocol must not have aborted or terminated through round  $r$ .<sup>10</sup>

**5.2.2. Claiming committed transactions / executing the protocol** : From times  $\tau = 1, \dots, \ell + 1$ , every player  $i$  does the following:

1. Read the ledger's state and compute the state of the protocol  $\pi_{SH}$ , given the messages posted by the participants.
2. If the protocol has not reached an aborted or terminated state given the current message transcript, calculate  $i$ 's message for round  $\tau$  and post it in a "claiming" transaction's `aux` field.
3. If the protocol has aborted or terminated, then post transactions reclaiming the funds from commitment transactions that have not been claimed.

To intuitively frame this in a rational setting, if the  $c$  coins in each transaction are valuable enough to the participants, then each participant will prefer to participate honestly in the protocol rather than defect at any round. If every party participates honestly in the protocol, then they will each obtain the desired output and net 0 coins. However, if a player  $i$  completes the setup phase but does not honestly complete the protocol, then that player may still learn the MPC output but will be down  $(n - 1) \cdot c$  coins.

We will formalize this intuition and specifically apply this compensation framework to secret sharing.

<sup>9</sup>We are using the time-restriction feature of the ledger, as rounds can be specified using time intervals in this setting.

<sup>10</sup>Thus using the state-dependent condition feature of the ledger. This enforces correct participation in the protocol if the participants wish to claim the coins.

### 5.3. Secret sharing

Kiayias *et al.* [21] include a substantial amount of detail on setting up a clock for synchronizing actions taken by the participants, but we will take for granted, as in the classical secret sharing setting, that participants act in synchronous, discrete rounds. This allows us to view the players' participation in the compensation protocol as an extensive form game, as we did in Sections 3.1 and 4.2.

**5.3.1. Setup** In the secret-sharing-with-compensation setting, each player  $i$  participates in a setup phase where they may choose to make the initial commitment to pay  $c$  coins to each player  $j$  that sends  $i$ 's secret. After the setup phase, the possible actions are the same as in section 3.1, but the utilities must be revised to account for the values of the coins. As in Section 3.1, let  $\text{info}(r) := (s_1, \dots, s_n)$  where  $s_i = 1$  if person  $i$  learns the secret and  $s_i = 0$  otherwise. Also, let  $\text{profit}(r) := (p_1, \dots, p_n)$  where  $p_i$  is the net profit in coin player  $i$  finishes the run with. We describe the utilities as follows, where  $U_i(r)$  describes the utility of player  $i$  given run  $r$ :

$$\text{U-1. } U_i(r) = U_i(r') \text{ if } \text{info}(r) = \text{info}(r') \text{ and } \text{profit}_i(r) = \text{profit}_i(r')$$

$$\text{U-2. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) > \text{info}_i(r')$$

$$\text{U-3. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) = \text{info}_i(r') \text{ and } \text{profit}_i(r) > \text{profit}_i(r')$$

$$\text{U-4. } U_i(r) > U_i(r') \text{ if } \text{info}_i(r) = \text{info}_i(r'), \text{profit}_i(r) \geq \text{profit}_i(r'), \text{info}_j(r) \leq \text{info}_j(r') \\ \text{for all } j \neq i, \text{ and there exists some } j \text{ such that } \text{info}_j(r) < \text{info}_j(r')$$

In this sense, each person  $i$  values first that they learn the secret, second that they maximize their net coin profit, and third that fewer other people learn the secret.<sup>11</sup> We'll see that this utility assumption is actually a fairly strong assumption – it leads us to be fairly demanding as to how much each player must value receiving  $c$  coins – but these assumptions allow us to find a positive result for having a secret sharing protocol that is (weakly) dominant-strategy-honest for rational players.

#### 5.3.2. Analysis

**Theorem 5.1.** *If the players' utilities satisfy U-1 – U-4, then if the secret sharing procedure has a commonly known bound of  $\ell$ , the only strategies that survive iterated deletion of weakly dominated strategies in every subgame are strategies in which each player  $i$  opts-in to the initial setup commitment and sends its secret share to all players  $j \neq i$  before the  $\ell$  rounds are up, and no player net gains or loses any coins.*

*Proof.* We again suppose that the game tree has  $\ell$  levels, where  $\ell$  is the commonly known bound on running time. Let us consider an arbitrary player  $i$  in an information set in which the set  $\mathcal{P}_{-i} = \{j \text{ such that } i \text{ has not sent its secret to } j\}$  is nonempty, and we are in level  $\ell$  of the tree (the last round). No action player  $i$  takes at this level affects whether or not player  $i$  learns the secret, so player  $i$  maximizes his utility by maximizing his net coin profit. (We have from our definition of the players' utility functions that the players value gains in number of coins over preventing other players from learning the secret.) The unique coin-profit-maximizing action to take at this information set is for player  $i$  to send his secret to every member of  $\mathcal{P}_{-i}$ . Thus for any player that

<sup>11</sup>Player  $i$  does not directly care about the net coin profit of the other players, although as the total net coin profit among all the participants sums to 0, player  $i$  receiving more coins is directly linked to the other players receiving fewer coins.

reaches round  $\ell$  without having sent her secret to every other player, it is clearly weakly dominant to send her secret to the rest of the players in round  $\ell$ .

We must also show that strategies in which  $\mathcal{P}_{-i} = \emptyset$  by algorithm termination are not weakly dominated for player  $i$ , including the fact that one such "all-send" strategy does not dominate any other "all-send" strategy. Let us consider any two arbitrary "all-send" strategies  $\sigma_i, \sigma'_i$ . These strategies must differ at some round  $r$  before the last round  $\ell$ . Then there exists strategy profiles for the rest of the players  $\sigma_{-i}, \sigma'_{-i}$  such that  $U_i(\sigma_i, \sigma_{-i}) > U_i(\sigma'_i, \sigma_{-i})$  and  $U_i(\sigma'_i, \sigma'_{-i}) > U_i(\sigma_i, \sigma'_{-i})$ . In these strategy profiles, all the other players do not send any messages until round  $r$  has passed. After round  $r$ , the players playing  $\sigma_{-i}$  will all send their shares if they are playing against  $\sigma_i$ , while the players playing  $\sigma'_{-i}$  will send their shares if they are playing against  $\sigma'_i$ . Then no "all-send" strategy is weakly dominated by another "all-send" strategy. If we extend this backwards before level  $\ell$ , we observe that because it is possible to reach an "all-send" on-path strategy starting from any information set of any level on the tree before level  $\ell$ , we can see that all "all-send" strategies survive  $\ell$  rounds of iterative deletion of weakly dominant strategies.

Let us now consider whether the strategy of opting-out at the setup phase is weakly dominated. We easily find opposing strategy profiles  $\sigma_{-i}$  for which an "all-send" strategy is higher than the utility derived from opting out at setup (e.g. all other players opt in at setup but do not fulfill any of their commitments to send messages, giving player  $i$   $(n-1)c$  coins), so "all-send" strategies are not weakly dominated by opting out at setup in any iteration.

We have already iteratively deleted all strategies that are not either "all-send" strategies or consist of opting-out at setup. We immediately have under this set of strategies, all players end the protocol with net zero coins. Then we consider three cases:

1. At least  $t$  players opt in during setup. Then none of the players who opted in benefit from switching to opting out (as they would no longer learn the secret), while players who did not opt in would benefit from switching to opting in (as they would then learn the secret).
2. Exactly  $t-1$  opt in during setup. Then the players who opted in are indifferent between their current strategy or switching to opting out, as either way no one learns the secret. The players who opted out would benefit from switching to opting in, as they would then learn the secret.
3. No greater than  $t-2$  players opt in during setup. Then all players are indifferent between opting in or opting out, as either way no one learns the secret.

Thus in the final iteration of deleting weakly dominated strategies, "all-send" strategies weakly dominate opting out at setup. Then strategies that opt in at setup and send one's secret to all participants are the only strategies that survive iterated deletion of weakly dominated strategies. Any strategy profile composed of only "all-send" strategies is a Nash equilibrium.  $\square$

We can additionally observe that it is sufficient to set  $\ell = 1$  so that all players will send all other players their secret shares in one round.

#### 5.4. Notes on coin valuations

While U-1 – U-4 are sufficient to imply that "all-send" strategies are the only strategies surviving iterated deletion of weakly dominated strategies, it is worth noting that finding a transaction amount  $c$  that allows these utilities to be satisfied in practice is nontrivial. For simplicity of demonstrating this point, in this section we assume that all players assign the same marginal utility to receiving  $c$

coins, regardless of the amount of coins they already have and the set of players that have learned the secret. (We will use the value  $c$  to denote this marginal utility.)

Let us consider an arbitrary player  $i$  with the following payoffs for protocol outcomes:

- All  $n$  players learning the secret:  $U_H$
- Only player  $i$  receiving output:  $U_y$
- No one receives output:  $U_0$

In keeping with the secret-sharing utility setting described previously, we have that  $U_y > U_H > U_0$ .

Let us consider the case that only  $t$  players have opted in at setup phase: we effectively have an  $n$ -out-of- $n$  secret sharing with compensation setting. Then we have that, assuming the other players participate honestly in the protocol, player  $i$  receives payoff  $U_H$ , while if player  $i$  opts in but does not send his secret to anyone, he receives  $U_y - (n - 1)c$ . Then in order for the protocol to be dominant-strategy-honest for  $i$ , we need that

$$\begin{aligned} U_H &\geq U_y - (n - 1)c \\ \implies c &\geq \frac{U_y - U_H}{(n - 1)} \end{aligned}$$

This result seems fairly intuitive: the coin payment that  $i$  receives for sending the necessary input to  $j$  must be at least as great as the utility  $i$  assigns to denying  $j$  the MPC output. Then this payment-commitment mechanism can be made dominant-strategy-honest simply by making  $c$  sufficiently high that this inequality holds true for all players.

This is nontrivial! Determining whether a given value  $c$  is sufficient in this scenario seems to necessitate determining which player assigns the highest value to  $U_y - U_H$  – or at least determining what the highest value is among the players. If the players wish to keep these values private, this appears to be equivalent to solving Yao’s Millionaires’ Problem... which is a problem solved with MPC. Even if the players are willing for these values not to be private, the player with the highest value for  $U_y - U_H$  has incentive to lie by declaring a lower value, and it’s not clear how a mechanism for eliciting the true value for this would be made compatible with this protocol.

## 6. Conclusion

In this work, we have studied various frameworks for  $t$ -out-of- $n$  secret sharing under rational players, as a methodology to replace trusted mediators in computing correlated equilibria. In particular, we have presented Halpern and Teague’s [18] results, in terms of showing the impossibility of classic deterministic secret sharing under iterated deletion of weakly dominated strategies in every subgame. We extend these results to demonstrate that it is similarly impossible to obtain deterministic secret sharing under a gradual release framework. However, we show that under a compensation framework, players are rationally motivated to carry out the protocol.

In terms of directions for future research, we have presented in Section 5.4 some key issues regarding calculating sufficient coin amounts to successfully carry out a compensation scheme; resolving these issues will be necessary to actually implement a compensation framework among rational players. Moreover, although we have obtained a positive result towards secret sharing, this does not necessarily directly extend to rational multiparty computation schemes; it would be

interesting to analyze which multiparty computation schemes can be rationally carried out assuming that secret sharing is accomplished using a compensation framework.

Finally, it is important to note that our results use a specific version of iterated deletion of weakly dominated strategies, in that we restrict ordering to a subgame hierarchy. In general, different results may be obtained by deleting weakly dominated strategies in different orderings, and it would be interesting to extend our results in Sections 4.2 and 5.3 to iterated deletion of weakly dominated strategies without a restriction on order.

## **7. Acknowledgements**

We would like to thank Professor Matt Weinberg, for his invaluable help and guidance in this work and throughout COS 521.<sup>12</sup>

---

<sup>12</sup>And for putting up with us.

## 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 *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 53–62, New York, NY, USA, 2006. ACM.
- [2] G. Asharov and Y. Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *J. Cryptol.*, 30(1):58–151, Jan. 2017.
- [3] C. Asmuth and J. Bloom. A modular approach to key safeguarding. *IEEE Trans. Inf. Theor.*, 29(2):208–210, Sept. 2006.
- [4] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.
- [5] D. Beaver and S. Goldwasser. Multiparty computation with faulty majority. In *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '89, pages 589–590, London, UK, UK, 1990. Springer-Verlag.
- [6] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. ACM.
- [7] G. R. Blakley. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on (AFIPS)*, volume 00, page 313, 12 1899.
- [8] M. Blum. How to exchange (secret) keys. *ACM Trans. Comput. Syst.*, 1(2):175–193, 1983.
- [9] D. Boneh. The decision diffie-hellman problem. In *Proceedings of the Third International Symposium on Algorithmic Number Theory*, ANTS-III, pages 48–63, London, UK, UK, 1998. Springer-Verlag.
- [10] D. Boneh and M. Naor. Timed commitments. In M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000: 20th Annual International Cryptology Conference Santa Barbara, California, USA, August 20–24, 2000 Proceedings*, pages 236–254. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
- [11] R. Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 364–369, New York, NY, USA, 1986. ACM.
- [12] I. B. Damgård. Practical and provably secure release of a secret and exchange of signatures. *Journal of Cryptology*, 8(4):201–222, Sep 1995.
- [13] Y. Dodis, S. Halevi, and T. Rabin. A cryptographic solution to a game theoretic problem. In M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000: 20th Annual International Cryptology Conference Santa Barbara, California, USA, August 20–24, 2000 Proceedings*, pages 112–130. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
- [14] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, June 1985.
- [15] J. A. Garay, P. D. MacKenzie, and K. Yang. Efficient and secure multi-party computation with faulty majority and complete fairness. *IACR Cryptology ePrint Archive*, 2004:9, 2004.
- [16] S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell. Complete fairness in secure two-party computation. *J. ACM*, 58(6):24:1–24:37, Dec. 2011.
- [17] S. D. Gordon and J. Katz. Rational secret sharing, revisited. In R. D. Prisco and M. Yung, editors, *Security and Cryptography for Networks: 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006. Proceedings*, pages 229–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [18] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: Extended abstract. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 623–632, New York, NY, USA, 2004. ACM.
- [19] S. Izmalkov, M. Lepinski, and S. Micali. Verifiably secure devices. In R. Canetti, editor, *Theory of Cryptography: Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. Proceedings*, pages 273–301. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [20] S. Izmalkov, S. Micali, and M. Lepinski. Rational secure computation and ideal mechanism design. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 585–594, Oct 2005.
- [21] A. Kiayias, H.-S. Zhou, and V. Zikas. Fair and robust multi-party computation using a global transaction ledger. In *Proceedings of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666*, pages 705–734, New York, NY, USA, 2016. Springer-Verlag New York, Inc.
- [22] M. Lepinski, S. Micali, and abhi shelat. Collusion-free protocols. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 543–552, New York, NY, USA, 2005. ACM.
- [23] M. Luby, S. Micali, and C. Rackoff. How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 11–22, Nov 1983.
- [24] A. Lysyanskaya and N. Triandopoulos. Rationality and adversarial behavior in multi-party computation. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology*, CRYPTO'06, pages 180–197, Berlin, Heidelberg, 2006. Springer-Verlag.

- [25] M. Mignotte. How to share a secret. In *Proceedings of the 1982 Conference on Cryptography*, pages 371–375, Berlin, Heidelberg, 1983. Springer-Verlag.
- [26] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Boston MA, 1994.
- [27] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
- [28] Y. Shoham and M. Tennenholtz. Non-cooperative computation: Boolean functions with correctness and exclusivity. *Theoretical Computer Science*, 343(1):97–113, 2005. Game Theory Meets Theoretical Computer Science.
- [29] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.