# Fair Computation with Rational Players*

Amos Beimel† Adam Groce‡ Jonathan Katz§ Ilan Orlov†

**Abstract**

We consider the problem of *fair* multiparty computation, where fairness means (informally) that all parties should learn the correct output. A seminal result of Cleve (STOC 1986) shows that fairness is, in general, impossible to achieve if a majority of the parties is *malicious*. Here, we treat all parties as *rational* and seek to understand what can be done.

Asharov et al. (Eurocrypt 2011) showed impossibility of rational fair computation in the two-party setting, for a particular function and a particular choice of utilities. We observe, however, that in their setting the parties have no strict incentive to compute the function *even in an ideal world where fairness is guaranteed*. Revisiting the problem, we show that rational fair computation *is* possible, for arbitrary functions, as long as the parties have a strict incentive to compute the function in an ideal world where fairness is guaranteed. Our results extend to more general utility functions that do not directly correspond to fairness, as well as to the multi-party setting. Our work thus shows a new setting in which game-theoretic considerations can be used to circumvent a cryptographic impossibility result.

## 1 Introduction

Cryptography and game theory are both concerned with understanding interactions between mutually distrusting parties with potentially conflicting interests. Cryptography typically adopts a "worst case" viewpoint; that is, cryptographic protocols are designed to protect the interests of each party against *arbitrary* (i.e., malicious) behavior of the other parties. The game-theoretic perspective, however, views parties as being *rational*; game-theoretic protocols, therefore, only need to protect against rational deviations by other parties.

Significant effort has recently been devoted to bridging cryptography and game theory; see [12, 28] for surveys. This work has tended to focus on two general sets of questions:

**Using cryptographic protocols to implement games** (e.g., [10, 13, 6, 11, 32, 1, 25, 24]). Given a game played by parties relying on an external trusted entity (a *mediator*), when can the mediator be replaced by a cryptographic protocol executed by the parties themselves?

**Applying game theory to cryptographic protocols** (e.g., [23, 28, 19, 33, 1, 20, 21]). What game-theoretic definitions are appropriate for computationally bounded, rational parties executing

*This is the full version of [22].

†Department of Computer Science, Ben Gurion University. Research supported by ISF grant #938/09. Email: {amos.beimel,ilanorlov}@gmail.com.

‡Department of Mathematics, Reed College. Portions of this work were done while at the University of Maryland. Email: agroce@reed.edu.

§Department of Computer Science, University of Maryland. Research supported by NSF awards #0830464, #0964541, #1111599, and #1223623. Email: jkatz@cs.umd.edu.

a protocol? Can impossibility results in the cryptographic setting be circumvented if we are willing to take a game-theoretic approach?

Here, we turn our attention to the question of *fair computation* in a rational setting, where fairness means that all parties should learn the value of some function $f$ evaluated on the parties' inputs. Following the work of Asharov et al. [4] (see further below), our goal is to understand when fairness is achievable by rational parties running some cryptographic protocol, without the aid of any external trusted entity. Our work touches on both the settings outlined above. Our motivation was to circumvent the strong impossibility result of Cleve [9] for fair computation without an honest majority by investigating a relaxed model where parties are rational but not malicious. In this sense, our work can be viewed as a generalization of the line of work on rational *secret sharing* [23, 19, 33, 1, 29, 30, 37, 35, 5, 14], which can be viewed as a special case of fair computation for a specific function with parties' inputs provided by a trusted dealer. It is also possible to view rational fair computation from a different perspective. Specifically, consider a natural "fairness game" involving a trusted mediator who computes a function $f$ on behalf of some parties (and gives all parties the result), and where parties can choose whether or not to participate and, if so, what input to send to the mediator. One can then ask whether there exists a real-world protocol (replacing the mediator) that preserves equilibria of the original mediated game. Our work demonstrates a close connection between these complementary viewpoints.

In addition to the above, we also consider more general utilities that do not directly correspond to fairness. Here, too, our goal is to understand when there exists a protocol such that running the protocol is a game-theoretic equilibrium with respect to the given utility function(s).

### 1.1 Our Results

We begin by discussing the two-party case where utilities correspond naturally to the problem of fairness. We then describe our more general results that include this setting as a special case.

**Fairness in the two-party setting.** Consider first the setting studied by Asharov et al. [4]. Here, there are two parties $P_1$ and $P_2$ who wish to compute a function $f$ of their inputs $x_1$ and $x_2$, where the joint distribution of $x_1$ and $x_2$ is common knowledge. (Asharov et al. assume that $x_1, x_2$ are uniform and independent but we allow for arbitrary distributions.) Furthermore, as in work on rational secret sharing, assume parties' utilities are such that each party prefers to learn the correct answer $f(x_1, x_2)$ and otherwise prefers that the other party outputs an *incorrect* answer. Informally, a cryptographic protocol computing $f$ is *rationally fair* if having both parties run the protocol is a (computational, Bayesian) Nash equilibrium with respect to fail-stop deviations.[1]

Asharov et al. show a negative result in this context: they give a function $f$, a distribution on the inputs, and a specific set of utilities for which there is *no* fair protocol computing $f$ with correctness better than $1/2$. They also show that correctness $1/2$ *can* be achieved (for that function, distribution, and utilities), but their work seemed to suggest that the power of rational fair computation is relatively limited.

Looking closely at their impossibility result, we observe that for their specific choices of $f$, the input distribution, and the utilities, *the parties have no strict incentive to run any protocol at all.* Namely, the utility each party obtains by running *any* protocol that correctly (and fairly) computes $f$ is equal to the expected utility that each party obtains if it simply guesses the input

[1] Later we will consider Byzantine deviations as well, but we assume fail-stop deviations here for simplicity.

of the other party and computes the function on its own (without any interaction).[2] In other words, *even in an ideal world* with a trusted mediator computing $f$ with complete fairness, the parties would be indifferent between using the trusted mediator or not. In game-theoretic terms, computing $f$ (even in an ideal world) is not a *strict* Nash equilibrium. If running a (real-world) protocol incurs any cost at all, there is thus little hope that the parties will prefer to run *any* protocol for computing $f$.

Asharov et al. rule out rational fair computation for a *specific* function, a *specific* input distribution, and a *specific* set of utilities. Are there any settings where rational fair computation (with complete correctness) *is* possible? Assuming the existence of secure oblivious transfer, we show:

**Theorem** (Informal) *Fix $f$, a distribution on inputs, and utility functions such that computing $f$ in the ideal world (with complete fairness) is a* **strict** *Nash equilibrium for both parties. Then, for the same input distribution and utility functions, there is a protocol $\Pi$ for computing $f$ (where correctness holds with all but negligible probability) such that following $\Pi$ is a computational Nash equilibrium. This holds in both the fail-stop and Byzantine settings.*

In addition to the fact that we show a positive result, our work goes beyond the setting considered in [4] in several respects: we handle arbitrary (deterministic) functions where parties receive possibly different outputs, and treat arbitrary distributions over the parties' inputs. (In [4], only single-output functions and independent, uniform input distributions were considered.) Moreover, we also treat the *Byzantine* setting where, in particular, parties have the option of changing their inputs; Asharov et al. [4] only treat the fail-stop case.

**The multiparty setting, and general utilities.** The preliminary version of our paper [22] only considered the two-party setting. Here, we also generalize the above to the multi-party setting, where there are $k$ parties and up to $t$ of them may be colluding. We first observe that rational fairness[3] is easy to achieve if completely fair computation for $t$ malicious parties is possible, i.e., when $t < k/2$ if a broadcast channel is available, and when $t < k/3$ otherwise. This is because no matter what the $t$ parties in the coalition do they will be unable to prevent the rest of the parties—assuming they follow the protocol—from learning the correct output. Thus, the problem is only interesting when $t, k$ are such that completely fair computation is not possible.

A next question is: what set of utilities best models fairness in the multi-party setting? Should the utility of a party in a coalition depend only on whether that party learns the correct answer, or also on whether *every* member of the coalition learns the correct answer? Similarly, does it matter whether no party outside the coalition learns the correct answer, or whether even one party outside the coalition fails to do so? Rather than choose arbitrarily among these options, we make *no* assumptions on the utility functions of the parties. Each party can have a different utility function, and in fact these utilities need not correspond to a notion of fairness at all. We show:

**Theorem** (Informal) *Fix $f$, a distribution on inputs, and utility functions such that computing $f$ in the ideal world is a* **strict** *Nash equilibrium for all coalitions of size at most $t$. Then, for the same input distribution and utility functions, there is a protocol $\Pi$ for computing $f$ (where correctness*

[2] Specifically, using the input distributions and utility functions from [4], a party's utility if both parties (run some protocol and) output the correct answer is 0, whereas if both parties guess then each party is (independently) correct with probability 1/2 and so the expected utility of each party is $\frac{1}{4} \cdot 1 + \frac{1}{4} \cdot (-1) + \frac{1}{2} \cdot 0 = 0$.

[3] We stress here that we consider only computational Nash equilibria, not stronger equilibrium notions. Our notion of rational fairness also allows the colluding parties to arbitrarily change their inputs.

*holds with all but negligible probability) such that following* $\Pi$ *is a computational Nash equilibrium for coalitions of size at most* $t$*. This holds in both the fail-stop and Byzantine settings.*

Note that this subsumes the result in the two-party case described earlier.

### 1.2 Other Related Work

The most relevant prior work is that of Asharov et al. [4], already discussed above. Here we merely add that their main motivation was to develop formal definitions of various cryptographic goals in a game-theoretic context, with fairness being only one example. Their paper takes an important step toward that goal.

As observed earlier, work on rational secret sharing [23, 19, 33, 1, 29, 30, 37, 35, 5, 14] can be viewed as a special case of fair secure computation, where the function being computed is the reconstruction function of the secret-sharing scheme being used, the parties' inputs are generated (and authenticated) by a dealer, and a specific class of utilities is assumed. Thus, in certain settings, our results give rational secret-sharing protocols where following the protocol is a (computational) Nash equilibrium. Most of the work on rational secret sharing, however, has focused on achieving *stronger* equilibrium notions, something we leave for future work.

An analogue of our results is given by Izmalkov et al. [31, 32, 27, 26] who, essentially, also show protocols for rational fair computation of a function whenever parties would prefer to compute that function in the ideal world. The main difference is that we assume standard communication channels, whereas the protocols of Izmalkov et al. require strong *physical* assumptions such as secure envelopes and ballot boxes.

There has recently been a significant amount of work on fairness in the cryptographic setting, showing functions that can be computed with complete fairness [16, 17, 2, 34, 3] and exploring various notions of partial fairness (see [18, 7] and references therein). The class of functions that are known to be computable with complete fairness is fairly limited; partial fairness and rational fairness are incomparable notions.

## 2 Model and Definitions

Given a deterministic function $f : X_1 \times \cdots \times X_k \to Y_1 \times \cdots \times Y_k$, we let $f_i$ denote the restriction of $f$ to its $i$th output. We consider two settings in which $k$ parties $P_1, \ldots, P_k$ wish to compute $f$ on their respective inputs $x_1, \ldots, x_k$, with $P_i$ receiving $f_i(x_1, \ldots, x_k)$: an *ideal-world* computation of $f$ using a trusted party, and a *real-world* computation of $f$ using some protocol $\Pi$. In each setting, the inputs $x_1, \ldots, x_k$ are chosen according to some joint probability distribution $D$ known to all parties, and in each setting we consider both fail-stop and Byzantine strategies.

We consider general utility functions defined over the (true) inputs[4] and (actual) outputs[5] of the parties. (This encompasses the case considered in [4, 22], where utilities are specifically assumed to model fairness.) We assume without loss of generality that utilities are nonnegative.

We consider potential deviations by *coalitions* of parties, with single-player deviations as a special case. If $\mathcal{C} \subset \{1, \ldots, k\}$ is a coalition, then $\overline{\mathcal{C}} \stackrel{\text{def}}{=} \{1, \ldots, k\} \setminus \mathcal{C}$. We let $x_{\mathcal{C}}$ (resp., $y_{\mathcal{C}}$) represent

[4] Parties may send any value to the trusted party (resp., use any value when running $\Pi$), but their true inputs are those given to them at the outset of the game.

[5] Parties may choose their actual output arbitrarily, and need not output the value given to them by the trusted party (resp., the value they obtain from running $\Pi$).

the inputs (resp., outputs) of the members of $\mathcal{C}$ and, for $i \in \mathcal{C}$, let $x_{\mathcal{C}}[i]$ (resp., $y_{\mathcal{C}}[i]$) denote the input (resp., output) of $P_i$. The utility of a coalition of parties is simply the sum of the utilities of its members. Following the cryptographic convention, we view a coalition of parties as being under the control of some central entity who coordinates the actions of the members of the coalition. We assume that the members of the coalition are able to freely communicate out-of-band, and so in particular they can share their inputs at the outset of the game.

### 2.1 Execution in the Ideal World

Our ideal world includes a trusted party who computes $f$ with complete fairness. This defines a natural game that proceeds as follows:

1. Inputs $x_1, \ldots, x_k$ are sampled according to a joint probability distribution $D$, and $x_i$ is then given to $P_i$.

2. Each party sends a value to the trusted party. We also allow parties to send a special value $\perp$ denoting an abort. Let $x'_i$ denote the value sent by $P_i$.

3. If any $x'_i = \perp$, the trusted party sends $\perp$ to all parties. Otherwise, the trusted party sends $f_i(x'_1, \ldots, x'_k)$ to each party $P_i$.

4. Each party $P_i$ outputs some value $y_i$ which need not be equal to the value it received from the trusted party.

In the *fail-stop* setting, we restrict $x'_i \in \{x_i, \perp\}$. In the *Byzantine* setting we allow $x'_i$ to be arbitrary.

The "desired" play in this game is for each party to send its input to the trusted party, and then output the value returned by the trusted party. To fully define an honest strategy, however, we must specify what each party does for every possible value (including $\perp$) it receives from the trusted party. Let $W_i$ denote a function from inputs to a distribution on outputs. We formally define strategy $(\mathsf{cooperate}, W_i)$ for $P_i$ as follows:

> $P_i$ sends its input $x_i$ to the trusted party. If the trusted party returns anything other than $\perp$, then $P_i$ outputs that value. If instead $\perp$ is returned, then $P_i$ generates output according to the distribution $W_i(x_i)$.

As long as all parties follow honest strategies, the distributions $W_i$ are irrelevant (as they are never used). They are important, however, insofar as they serve as *threats* in case of an abort by other parties: namely, $P_1$ (for example) knows that if he aborts then every other $P_i$ will determine its own output according to $W_i(x_i)$, and so $P_1$ must take this into account when deciding whether to abort. The situation in which each party $P_i$ plays $(\mathsf{cooperate}, W_i)$ is a *$t$-resilient, Bayesian, strict Nash equilibrium* if the following two conditions hold:

1. For every coalition $\mathcal{C}$ of size at most $t$, every (allowed[6]) deviation by the members of $\mathcal{C}$ does not increase the expected utility of $\mathcal{C}$.

2. Moreover, for every coalition $\mathcal{C}$ of size at most $t$, every (allowed) deviation by the members of $\mathcal{C}$ that has $x'_{\mathcal{C}} \neq x_{\mathcal{C}}$ with nonzero probability results in *strictly lower* expected utility for $\mathcal{C}$.

[6] I.e., in the fail-stop case the only allowed deviation is aborting, whereas in the Byzantine case parties are allowed to send arbitrary inputs. In either case deviating parties may determine their outputs arbitrarily.

The second condition means that the members of any coalition have a strict incentive to send their true inputs to the trusted party. The first condition implies that, having done so, these parties have no incentive to output anything other than what they received from the trusted party.

**Definition 1.** *Fix $f$, a distribution $D$, and utility functions $\{U_i\}_{i=1}^k$. We say these are $t$-*incentive compatible in the fail-stop (resp., Byzantine) setting *if there exist $\{W_i\}_{i=1}^k$ such that the strategy profile $\Big((\mathsf{cooperate}, W_1), \ldots, (\mathsf{cooperate}, W_k)\Big)$ is a $t$-resilient, Bayesian, strict Nash equilibrium in the ideal-world game defined above.*

**The setting of Asharov et al. [4].** For completeness, we show that the (two-party) setting considered by Asharov et al. is *not* incentive compatible, even in the fail-stop setting. In [4] the utilities are such that (1) getting the correct answer while the other party outputs an incorrect answer gives utility 1; (2) getting an incorrect answer while the other party outputs the correct answer gives utility $-1$; and (3) any other outcome gives utility 0. Furthermore (cf. [4, Definition 4.6]), $f$ corresponds to boolean XOR, and the inputs for each party are chosen uniformly and independently. We claim that there is no choice of $W_1, W_2$ for which $\big((\mathsf{cooperate}, W_1), (\mathsf{cooperate}, W_2)\big)$ is a 1-resilient, Bayesian, strict Nash equilibrium. To see this, fix $W_1, W_2$ and note that playing $\big((\mathsf{cooperate}, W_1), (\mathsf{cooperate}, W_2)\big)$ gives utility 0 to both parties. On the other hand, if $P_1$ aborts and outputs a random bit, then regardless of the guessing strategy $W_2$ employed by $P_2$, the parties $P_1$ and $P_2$ are each correct with independent probability $1/2$ and so the expected utility of $P_1$ remains 0. This shows an allowed deviation by $P_1$ that does not result in lower expected utility for $P_1$.

In contrast, if the utilities are modified so that when both parties get the correct answer they each obtain utility $1/2$ (and everything else is unchanged), then the setting *is* incentive compatible in the fail-stop setting. To see this, let $W_1, W_2$ be the uniform distribution (regardless of the input). Playing $\big((\mathsf{cooperate}, W_1), (\mathsf{cooperate}, W_2)\big)$ gives utility $1/2$ to both parties. If, on the other hand, $P_1$ ever aborts on some input then—no matter how $P_1$ determines its output—$P_1$ and $P_2$ are each correct with independent probability $1/2$. The expected utility of deviating is $1/8$, which is strictly smaller than $1/2$; thus, we have a Bayesian *strict* Nash equilibrium. Our results imply that a rational (fair) protocol *can* be constructed for this setting.

## 2.2 Execution in the Real World

In the real world there is no trusted party, and the players instead must communication in order to compute $f$. We thus have a real-world game in which inputs $x_1, \ldots, x_k$ are jointly sampled according to $D$, input $x_i$ is given to $P_i$, and the parties execute some strategy (i.e., run some protocol) and then decide on their respective outputs.

The goal is to construct a protocol $\Pi$ such that running the protocol is a (computational) Nash equilibrium. In designing the protocol, we assume the existence of a secure communication channel between each pair of parties, as well as a broadcast channel. (These could both be realized using standard cryptographic techniques.) We assume that communication occurs in synchronous rounds, but allow *rushing* so that in any given round a deviating coalition obtains the messages sent to it by other parties before sending its own messages for that round.

The running times of the parties, as well as the protocol itself, are parameterized in terms of a security parameter $n$; however, the function $f$ as well as the parties' utilities are fixed and independent of $n$. We assume all parties run in probabilistic polynomial-time (in $n$), and only consider protocols where correctness holds with all but negligible probability (in $n$).

As in the ideal world, we again consider two types of deviations. In the *fail-stop* setting, each party follows the protocol as directed except that it may choose to abort the protocol at any point. Upon aborting, a party may output whatever value it likes (and not necessarily the value prescribed by the protocol). Parties who follow the protocol decide on their output as prescribed by the protocol. We stress that in the fail-stop setting a party is assumed not to change its input when running the protocol. In contrast, in the *Byzantine* setting parties may behave arbitrarily (and, in particular, may run the protocol using a different input).

We now define what it means for $\Pi$ to induce a game-theoretic equilibrium. Our equilibrium notion of interest is (Bayesian) computational Nash equilibrium (see [28]). We say a protocol induces a *$t$-resilient computational Nash equilibrium* if any (allowed) deviation by a coalition $\mathcal{C}$ of at most $t$ probabilistic polynomial-time parties yields expected payoff at most negligibly more than what $\mathcal{C}$ can obtain by running the protocol honestly (and then outputting the value prescribed by the protocol); in other words, deviating from the protocol cannot increase the expected utility of $\mathcal{C}$ more than a negligible amount. (By *Bayesian* we simply mean that we take the expectation also over the possible inputs of the parties outside the coalition $\mathcal{C}$.) In our setting, then, we have the following definition of what it means for $\Pi$ to be a rational protocol (the definition is equivalent to a generalized version of the definition used by Asharov et al. [4, Definition 4.6]):

**Definition 2.** *Fix $f$, a distribution $D$, utilities for the parties, and a protocol $\Pi$ computing $f$. We say $\Pi$ is a $t$-*rational protocol *(with respect to these parameters)* in the fail-stop (resp., Byzantine) setting *if running $\Pi$ is a $t$-resilient, Bayesian, computational Nash equilibrium in the real-world game defined above.*

We consider computational Nash equilibria, rather than computational *strict* Nash equilibria, since the latter are notoriously difficult to define [14]; also, the goal of our work is only to construct real-world protocols that induce a Nash equilibrium. (We define strict Nash equilibria in the ideal world only because we use it for our results.) We stress that it only makes sense to speak of $\Pi$ being a rational protocol with regard to some specific input distribution and utilities; it is possible, e.g., for $\Pi$ to be rational for one set of utilities but not another.

# 3 Positive Results for Rational Computation

We show broad positive results for rational computation in both the fail-stop and Byzantine settings. Namely, we show that whenever computing the function honestly is a $t$-resilient, Bayesian, strict Nash equilibrium in the ideal world, then there exists a protocol $\Pi$ computing $f$ such that running $\Pi$ is a Bayesian computational Nash equilibrium in the real world.

Our protocols all share a common structure. As in prior work on fairness [16, 36, 18], our protocols have two stages. The first stage is a "pre-processing" step that relies on any protocol for (unfair) secure multi-party computation, and the second stage takes place in a sequence of $n$ iterations, where $n$ is the security parameter. In our work, the stages take the following form:

**First stage:**

1. A value $r^* \in \{1, \ldots\}$ is chosen according to a geometric distribution. This represents the iteration (unknown to the parties) in which all parties will learn the correct output.
2. Values $\{(t_1^r, \ldots, t_k^r)\}_{r=1}^n$ are chosen, with $t_i^r$ the value that $P_i$ should learn in iteration $r$. For $r \geq r^*$ we have $t_i^r = f_i(x_1, \ldots, x_k)$, while for $r < r^*$ the value $t_i^r$ depends on $P_i$'s input only (see below for details).

3. Each $t_I^r$ value is shared in a $k$-out-of-$k$ manner among all parties.

**Second stage:** In each iteration $r \in \{1, \ldots, n\}$, for every $i \in \{1, \ldots, k\}$ each party other than $P_i$ broadcasts its share of $t_i^r$, thus allowing (only) $P_i$ to reconstruct $t_i^r$. When the protocol ends (either through successful termination or an abort by the other party) each party $P_i$ outputs the most-recently-learned value $t_i^r$.

The key difference with respect to prior work is how we set the distribution of the $\{t_i^r\}$ for $r < r^*$. Here we use the assumption that $f, D$, and the utilities are incentive compatible, and thus there are "guessing strategies" $W_i(x_i)$ for the parties—in case the protocol is aborted—that are in equilibrium (see Section 2.1). We use exactly these distributions in our protocol.

### 3.1 The Fail-Stop Setting

We first present an analysis of the fail-stop setting, which already demonstrates the main issues. We say $W_i$ has *full support* if for every $x_i$ the distribution $W_i(x_i)$ puts nonzero probability on every element in the range of $f_i$. We begin by proving a technical lemma.

**Lemma 3.** *Fix a function $f$, a distribution $D$, and utilities for the parties that are $t$-incentive compatible in the fail-stop (resp., Byzantine) setting. Then there exist $\{W_i\}$* **with full support** *such that $\Big((\mathsf{cooperate}, W_1), \ldots, (\mathsf{cooperate}, W_k)\Big)$ is a $t$-resilient, Bayesian, strict Nash equilibrium in the fail-stop (resp., Byzantine) setting.*

**Proof** We focus on the fail-stop setting; the proof for the Byzantine setting is analogous. Incentive compatibility means that there exist distributions $\{W_i'\}$ such that the strategy vector $\Big((\mathsf{cooperate}, W_1'), \ \ldots, \ (\mathsf{cooperate}, W_k')\Big)$ is a $t$-resilient, Bayesian, strict Nash equilibrium. The $\{W_i'\}$ may not have full support, but we show that they can be modified so that they do.

Let $u_{\max}$ denote the highest possible utility a coalition of size at most $t$ can obtain. For some coalition $\mathcal{C}$ and inputs $x_{\mathcal{C}}$ for the members of this coalition, let $u_\perp(x_{\mathcal{C}})$ denote the maximum expected utility this coalition can obtain if it aborts; this expectation is taken over the distribution $D$ from which inputs are chosen, as well as the randomized strategies—determined by $\{W_i'\}$—that are used by the players outside of $\mathcal{C}$. Let $u(x_{\mathcal{C}})$ be the expected utility the coalition obtains if each member of the coalition sends its input to the trusted party and then outputs the value received from the trusted party. Let $u^*$ be the minimum value of $u(x_{\mathcal{C}}) - u_\perp(x_{\mathcal{C}})$, where this minimum is taken over all coalitions $\mathcal{C}$ of size at most $t$ and over all inputs $x_{\mathcal{C}}$. By definition of incentive compatibility, we must have $u^* > 0$.

Set $\epsilon = \frac{u^*}{2k \cdot u_{\max}} > 0$, and define $W_i(x_i)$ as follows: with probability $(1 - \epsilon)$ choose output according to $W_i'(x_i)$, and with probability $\epsilon$ output a uniform element from the range of $f_i$. Note that $W_i$ has full support. We claim that the strategy vector $\Big((\mathsf{cooperate}, W_1), \ldots, (\mathsf{cooperate}, W_k)\Big)$ is a $t$-resilient, Bayesian, strict Nash equilibrium. To see this, consider any coalition $\mathcal{C}$ of size at most $t$, and some set of inputs $x_{\mathcal{C}}$ for the members of this coalition. In case of an abort by any member of $\mathcal{C}$, note that the parties outside $\mathcal{C}$ all follow the original strategies $\{W_i'\}$ with probability at least $1 - \epsilon k$. Therefore, the maximum expected utility of $\mathcal{C}$ following an abort (assuming the players outside of $\mathcal{C}$ determine their outputs using the modified strategies $\{W_i\}$) is at most

$$u_\perp(x_{\mathcal{C}}) + \epsilon k \cdot u_{\max} < u_\perp(x_{\mathcal{C}}) + u^* = u(x_{\mathcal{C}}).$$

**Functionality** ShareGen

**Inputs:** ShareGen takes as input a value $x_i$ from each $P_i$.

**Computation:** Proceed as follows:

1. If any $x_i$ input is invalid, then output $\perp$ to all parties.
2. Choose $r^*$ according to a geometric distribution with parameter $p$.
3. Set the values of $t_i^r$ for every $r \in \{1, \ldots, n\}$ and every $i \in \{1, \ldots, k\}$ as follows:
   - If $r < r^*$, choose $t_i^r \leftarrow W_i(x_i)$.
   - If $r \geq r^*$, set $t_i^r = f_i(x_1, \ldots, x_k)$.
4. For each $t_i^r$, choose values $s_{i,j}^r$ as random $k$-out-of-$k$ secret shares of $t_i^r$. I.e., the $\{s_{i,j}^r\}_{j=1}^k$ are chosen uniformly subject to $\bigoplus_{j=1}^k s_{i,j}^r = t_i^r$.

**Output:** Send $P_j$ the values $s_{i,j}^r$ for all $i$ and $r$.

Figure 1: Functionality ShareGen. The security parameter is $n$. This functionality is parameterized by a real number $p > 0$ to be determined later.

That is, aborting results in strictly lower utility than behaving honestly. □

We now formally state and prove our main theorem.

**Theorem 1.** *Fix a function $f$, a distribution $D$, and utilities for the parties. If these are $t$-incentive compatible in the fail-stop setting, then (assuming the existence of general secure multiparty computation for $t$ fail-stop adversaries) there exists a protocol $\Pi$ computing $f$ such that $\Pi$ is a $t$-rational protocol (with respect to the same distribution and utilities) in the fail-stop setting.*

**Proof** By definition of $t$-incentive compatibility, and using Lemma 3, there exist $\{W_i\}$ with full support for which the strategy profile $\Big((\mathsf{cooperate}, W_1), \ldots, (\mathsf{cooperate}, W_k)\Big)$ is a $t$-resilient, Bayesian, strict Nash equilibrium. We define a functionality ShareGen (cf. Figure 1) based on these $\{W_i\}$; this functionality is also parameterized by a constant $p > 0$ that we will set later. We define our protocol $\Pi$, that uses ShareGen as a building block, in Figure 2.

ShareGen starts by choosing a special iteration $r^*$ according to a geometric distribution with a parameter $p$ to be defined later. It then creates outputs for each party in each iteration. In iterations before $r^*$, the outputs are chosen independently using the distributions $\{W_i(x_i)\}$. From iteration $r^*$ on, the outputs are the actual value of the function. These output values are then shared in a $k$-out-of-$k$ manner, with shares sent to each player, to prevent the output values from being learned prematurely. Since $p$ is a constant, we have $r^* \leq n$ with overwhelming probability, and hence when $\Pi$ is run honestly all parties obtain the correct answer with all but negligible probability. (Alternately, one could modify ShareGen to enforce that $r^* \leq n$ always.)

Protocol $\Pi$ can use any multiparty computation protocol $\pi$ for computing ShareGen that is secure with unanimous abort [15] against $t$ fail-stop adversaries. Since we assume a fail-stop adversarial model, this means we can analyze $\Pi$ in a hybrid world where there is a trusted entity computing ShareGen *without fairness* but with agreement on abort. (This means that a deviating coalition may learn its output from ShareGen while all remaining parties receive $\perp$.) It is not hard to see (following [8]) that if $\Pi$ is a computational Nash equilibrium in this hybrid world, then so is $\Pi$ when executed in the real world (with a secure protocol implementing ShareGen).

**Protocol $\Pi$**

**Stage one:** Parties execute a secure protocol for computing ShareGen. This results in each party $P_j$ obtaining output $s^r_{i,j}$ for $1 \leq i \leq k$ and $1 \leq r \leq n$. (If ShareGen returns $\perp$, go to the output-determination phase.)

**Stage two:** There are $n$ iterations. In each iteration $r \in \{1, \ldots, n\}$ do:

- Each $P_j$ broadcasts $\{s^r_{i,j}\}_{i \neq j}$.
- If some party does not broadcast a value, go to the output-determination phase. Otherwise, each $P_j$ computes $t^r_j = \bigoplus_{i=1}^{k} s^r_{j,i}$.

**Output determination:** Each party $P_j$ determines its output as follows:

- If an abort occurs before $P_j$ has computed $t^1_j$, then $P_j$ chooses its output according to distribution $W_j(x_j)$.
- If an abort occurs at any other point, or the protocol completes successfully, then $P_i$ outputs the last $t^r_i$ value it computed.

Figure 2: Formal definition of our protocol.

Our goal is to show that there exists a $p > 0$ for which $\Pi$ (in the hybrid world discussed above) is a $t$-rational protocol with respect to the given function $f$, distribution $D$, and parties' utilities. Fix an arbitrary coalition $\mathcal{C}$ of size at most $t$, and an arbitrary input $x_{\mathcal{C}}$ for the parties in $\mathcal{C}$. We show that no fail-stop deviation by $\mathcal{C}$ can increase the expected utility of the coalition. (Because we are now in a hybrid world with access to an ideal functionality computing ShareGen, we no longer need to restrict attention to polynomial-time behavior, or consider negligible changes to the expected utility.)

Before continuing the analysis, we introduce two conceptual changes to the protocol that can only increase the expected utility of $\mathcal{C}$. First, at the outset of each iteration $r$ we first inform $\mathcal{C}$ whether $r > r^*$ or not. Second, if any party in $\mathcal{C}$ ever aborts in some iteration $r$ with $r \leq r^*$, then we inform the parties in $\mathcal{C}$ whether $r = r^*$ before those parties generate their output. (The decision to abort, however, cannot be changed.)

Note that once $r > r^*$, the coalition $\mathcal{C}$ cannot increase its expected utility by deviating from the protocol: although the parties in $\mathcal{C}$ have learned the correct outputs of the function, the parties outside $\mathcal{C}$ have also obtained their respective correct outputs in iteration $r^*$ ,and will only ever output those values from then on. Thus, this situation is analogous to the situation in the ideal world once all parties have been given their outputs by the trusted party. Since, by assumption, the parties in $\mathcal{C}$ have no incentive to deviate—namely, to output values other than the ones given to them by the trusted party—in the latter case, they have no incentive to deviate in the former case, either.

Let $u_{\max}$ denote the highest possible utility a coalition of size $t$ can obtain. Let $u(x_{\mathcal{C}})$ be the expected utility of the parties in $\mathcal{C}$ if they follow the protocol honestly when given inputs $x_{\mathcal{C}}$, and let $u_{\perp}(x_{\mathcal{C}})$ be the maximum expected utility the coalition can obtain (in the ideal world) if it aborts when given those inputs. Let $u^*$ be the minimum value of $u(x_{\mathcal{C}}) - u_{\perp}(x_{\mathcal{C}})$, where this minimum is taken over all coalitions $\mathcal{C}$ of size at most $t$ and all inputs $x_{\mathcal{C}}$; since we have a strict Nash equilibrium in the ideal world, $u^* > 0$.

Fix some inputs $x_{\mathcal{C}}$ for the parties in $\mathcal{C}$, and consider some iteration $r < n$ in which $r \leq r^*$. The parties in $\mathcal{C}$ learn their respective values $t^{\mathcal{C}}_r$ and can then decide whether to abort or not. If they

do not abort, and execute the protocol until the end, they receive expected utility $u(x_{\mathcal{C}})$. On the other hand, if they abort then with some probability $\alpha$ (which may depend on both $x_{\mathcal{C}}$ and $t^r_{\mathcal{C}}$) they are told that $r = r^*$, in which case they can potentially obtain utility $u_{\max}$; with probability $1-\alpha$, however, they are told that $r > r^*$ in which case they receive expected utility at most $u_\perp(x_{\mathcal{C}})$ (as $\mathcal{C}$ has no information beyond what it could compute from its own inputs). That is, the expected utility of aborting is at most

$$\alpha \cdot u_{\max} + (1-\alpha) \cdot u_\perp(x_{\mathcal{C}}).$$

Thus, as long as $\alpha \leq u^*/u_{\max}$, the expected utility of aborting is at most the expected utility of following the protocol. We now show that the parameter $p > 0$ can be set so that the stated bound on $\alpha$ holds (for all $x_{\mathcal{C}}$ and $t^r_{\mathcal{C}}$).

Let $q = \min_{x_{\mathcal{C}}, t_{\mathcal{C}}}\{\Pr[\forall P_i \in \mathcal{C} : W_i(x_i) = t_i]\}$. Since the $\{W_i\}$ have full support, $q > 0$. Now, for any coalition $\mathcal{C}$ given inputs $x_{\mathcal{C}}$ and observing outputs $t_{\mathcal{C}}$ in some iteration $r \leq r^*$, we have

$$\begin{aligned}
\alpha &\stackrel{\text{def}}{=} \Pr[r^* = r \mid t^r_{\mathcal{C}} = t_{\mathcal{C}} \wedge r^* \geq r] \\
&= \frac{\Pr[r^* = r \wedge t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* \geq r]}{\Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* \geq r]} \\
&= \frac{\Pr[r^* = r \mid r^* \geq r] \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* = r]}{\Pr[r^* = r \mid r^* \geq r] \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* = r] + \Pr[r^* > r \mid r^* \geq r] \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* > r]} \\
&= \frac{p \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* = r]}{p \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* = r] + (1-p) \cdot \Pr[t^r_{\mathcal{C}} = t_{\mathcal{C}} \mid r^* > r]} \\
&\leq \frac{p}{p + (1-p) \cdot q} \\
&= \frac{p}{p \cdot (1-q) + q} \leq \frac{p}{q}.
\end{aligned}$$

We thus see that by setting $p$ to a sufficiently small (positive) constant, we can ensure $\alpha \leq u^*/u_{\max}$ as required above.

Assuming $p$ is set as just described, the above shows that in any iteration $r < n$, the coalition $\mathcal{C}$ has no incentive to abort. The only remaining case to analyze is when $r = n$. In this case it would indeed be advantageous for $\mathcal{C}$ to abort when $r^* \geq n$; however, this occurs with only negligible probability and so does not impact the expected utility of $\mathcal{C}$ by more than a negligible amount. □

Although security notions beyond game-theoretic equilibria are not the focus of our work, we note that the protocol presented in the proof of the previous theorem is private in addition to being rational. That is, the parties learn the function output only, but nothing else regarding other parties' inputs.

### 3.2 The Byzantine Setting

The fail-stop setting already captures the main difficulties of the problem. We can handle the Byzantine setting by modifying the protocol from the previous section. Specifically, we first modify the ShareGen functionality so that it *authenticates* the shares given to each of the parties. The simplest way to handle this is to have ShareGen also generate public and private keys for a digital signature scheme, sign each share given to each party (along with the party's identifier and the iteration number), and output the public key to all parties. In the protocol itself, the parties should compute ShareGen using a sub-protocol that is secure (with unanimous abort) against $t$ *malicious*

parties. Moreover, all parties should now verify the signatures on all broadcast shares, and treat an invalid or missing signature as an abort. We omit the details, which are straightforward.

A proof of the following is completely analogous to the proof in the previous section:

**Theorem 2.** *If a function $f$, a distribution $D$, and utilities for the parties are $t$-incentive compatible in the Byzantine setting, then (assuming the existence of digital signatures and general secure multiparty computation for $t$ malicious adversaries) there exists a protocol $\Pi$ computing $f$ that is a $t$-rational protocol (with respect to the same distribution and utilities) in the Byzantine setting.*

# 4 Conclusions and Future Work

Given the stark impossibility results for fairness in a purely *malicious* context [9], it is natural to try to understand whether, or to what extent, fairness is achievable in a *rational* setting. Asharov et al. [4] gave a somewhat pessimistic answer to this question, as they show a specific case where rational fairness *cannot* be achieved (if correctness better than 1/2 is desired). Our work, in contrast, shows broad feasibility results for rational fairness: roughly, we show that whenever computing the function is a *strict* Nash equilibrium in the ideal world, then it is possible to construct a rational fair protocol computing the function in the real world.

Within the broader context of research at the intersection of game theory and cryptography, our result can be interpreted in two ways:

- Given a "fairness game" defined in an ideal world where there is a trusted entity (i.e., a "mediator") computing some function on behalf of the parties, a natural question to ask is when a game-theoretic equilibrium in the ideal world can be implemented via a real-world protocol. We do not provide a complete answer to this question, but we do show a partial characterization: roughly, whenever there is a *strict* Nash equilibrium in the ideal world, there is a protocol that induces a computational Nash equilibrium in the real world.
- We show a new setting in which cryptographic impossibility results can be circumvented by assuming rational behavior. Viewed in this light, our results can be seen as a generalization of work on rational secret sharing.

Our work suggests several interesting directions for future research. First, can positive results be shown even when our definition of a strict Nash equilibrium is relaxed? Or can a converse of our result be shown, at least under certain conditions? It will also be interesting to explore stronger game-theoretic solution concepts in the real world. We construct real-world protocols that induce a computational Nash equilibrium, but one could also aim to construct protocols satisfying some of the stronger equilibrium notions proposed, e.g., in [23, 29, 30, 14].

# 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 *25th Annual ACM Symp. on Principles of Distributed Computing (PODC)*, pages 53–62. ACM Press, 2006.

[2] G. Asharov. Towards characterizing complete fairness in secure two-party computation. In *9th Theory of Cryptography Conference—TCC 2014*, volume 8349 of *LNCS*, pages 291–316. Springer, 2014.

[3] G. Asharov, A. Beimel, N. Makriyannis, and E. Omri. Complete characterization of fairness in secure two-party computation of Boolean functions. In *9th Theory of Cryptography Conference—TCC 2015*, volume 9014 of *LNCS*. Springer, 2015.

[4] G. Asharov, R. Canetti, and C. Hazay. Towards a game theoretic view of secure computation. In *Advances in Cryptology—Eurocrypt 2011*, volume 6632 of *LNCS*, pages 426–445. Springer, 2011. Full version available at `http://eprint.iacr.org/2011/137`.

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

[6] I. Barany. Fair distribution protocols, or how the players replace fortune. *Mathematics of Operations Research*, 17:327–340, 1992.

[7] A. Beimel, Y. Lindell, E. Omri, and I. Orlov. $1/p$-Secure multiparty computation without honest majority and the best of both worlds. In *Advances in Cryptology—Crypto 2011*, volume 6841 of *LNCS*, pages 277–296. Springer, 2011.

[8] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[9] R. Cleve. Limits on the security of coin flips when half the processors are faulty. In *18th Annual ACM Symp. on Theory of Computing (STOC)*, pages 364–369. ACM Press, 1986.

[10] V.P. Crawford and J. Sobel. Strategic information transmission. *Econometrica*, 50:1431–1451, 1982.

[11] Y. Dodis, S. Halevi, and T. Rabin. A cryptographic solution to a game theoretic problem. In *Advances in Cryptology—Crypto 2000*, volume 1880 of *LNCS*, pages 112–130. Springer, 2000.

[12] Y. Dodis and T. Rabin. Cryptography and game theory. In N. Nisan, T. Roughgarden, É. Tardos, and V.V. Vazirani, editors, *Algorithmic Game Theory*, pages 181–207. Cambridge University Press, 2007.

[13] F. Forges. Universal mechanisms. *Econometrica*, 58:1342–1364, 1990.

[14] G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In *7th Theory of Cryptography Conference—TCC 2010*, volume 5978 of *LNCS*, pages 419–436. Springer, 2010.

[15] S. Goldwasser and Y. Lindell. Secure multi-party computation without agreement. *Journal of Cryptology*, 18(3):247–287, 2005.

[16] S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell. Complete fairness in secure two-party computation. *J. ACM*, 58(6), 2011.

[17] S. D. Gordon and J. Katz. Complete fairness in multi-party computation without an honest majority. In *6th Theory of Cryptography Conference—TCC 2009*, volume 5444 of *LNCS*, pages 19–35. Springer, 2009.

[18] S. D. Gordon and J. Katz. Partial fairness in secure two-party computation. *Journal of Cryptology*, 25(1):14–40, 2012.

[19] S. D. Gordon and Jonathan Katz. Rational secret sharing, revisited. In *5th Intl. Conf. on Security and Cryptography for Networks*, volume 4116 of *LNCS*, pages 229–241. Springer, 2006.

[20] R. Gradwohl. Rationality in the full-information model. In *7th Theory of Cryptography Conference—TCC 2010*, volume 5978 of *LNCS*, pages 401–418. Springer, 2010.

[21] R. Gradwohl, N. Livne, and A. Rosen. Sequential rationality in cryptographic protocols. In *51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 623–632. IEEE, 2010.

[22] A. Groce and J. Katz. Fair computation with rational players. In *Advances in Cryptology—Eurocrypt 2012*, volume 7237 of *LNCS*, pages 81–98. Springer, 2012.

[23] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation. In *36th Annual ACM Symp. on Theory of Computing (STOC)*, pages 623–632. ACM Press, 2004.

[24] P. Hubáček, J. Buus Nielsen, and A. Rosen. Limits on the power of cryptographic cheap talk. In *Advances in Cryptology—Crypto 2013, Part I*, volume 8042, pages 277–297. Springer, 2013.

[25] S. Izmalkov, M. Lepinski, and S. Micali. Perfect implementation. *Games and Economic Behavior*, 71(1):121–140, 2011. Available at `http://hdl.handle.net/1721.1/50634`.

[26] Sergei Izmalkov, Matt Lepinski, and Silvio Micali. Verifiably secure devices. In *5th Theory of Cryptography Conference—TCC 2008*, volume 4948 of *LNCS*, pages 273–301. Springer, 2008.

[27] Sergei Izmalkov, Silvio Micali, and Matt Lepinski. Rational secure computation and ideal mechanism design. In *46th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 585–595. IEEE, 2005. Full version available at `http://dspace.mit.edu/handle/1721.1/38208`.

[28] Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. In *5th Theory of Cryptography Conference—TCC 2008*, volume 4948 of *LNCS*, pages 251–272. Springer, 2008.

[29] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *5th Theory of Cryptography Conference—TCC 2008*, volume 4948 of *LNCS*, pages 320–339. Springer, 2008.

[30] Gillat Kol and Moni Naor. Games for exchanging information. In *40th Annual ACM Symp. on Theory of Computing (STOC)*, pages 423–432. ACM Press, 2008.

[31] Matt Lepinski, Silvio Micali, Chris Peikert, and Abhi Shelat. Completely fair SFE and coalition-safe cheap talk. In *23rd Annual ACM Symp. on Principles of Distributed Computing (PODC)*, pages 1–10. ACM Press, 2004.

[32] Matt Lepinski, Silvio Micali, and Abhi Shelat. Collusion-free protocols. In *37th Annual ACM Symp. on Theory of Computing (STOC)*, pages 543–552. ACM Press, 2005.

[33] Anna Lysyanskaya and Nikos Triandopoulos. Rationality and adversarial behavior in multi-party computation. In *Advances in Cryptology—Crypto 2006*, volume 4117 of *LNCS*, pages 180–197. Springer, 2006.

[34] N. Makriyannis. On the classification of finite Boolean functions up to fairness. In *9th International Conference on Security and Cryptography for Networks (SCN)*, volume 8642 of *Lecture Notes in Computer Science*, pages 135–154. Springer, 2014.

[35] S. Micali and A. Shelat. Truly rational secret sharing. In *6th Theory of Cryptography Conference—TCC 2009*, volume 5444 of *LNCS*, pages 54–71. Springer, 2009.

[36] T. Moran, M. Naor, and G. Segev. An optimally fair coin toss. In *6th Theory of Cryptography Conference—TCC 2009*, volume 5444 of *LNCS*, pages 1–18. Springer, 2009.

[37] S.J. Ong, D. Parkes, A. Rosen, and S. Vadhan. Fairness with an honest minority and a rational majority. In *6th Theory of Cryptography Conference—TCC 2009*, volume 5444 of *LNCS*, pages 36–53. Springer, 2009.