Utility Dependence in Correct and Fair Rational Secret Sharing*

Gilad Asharov† Yehuda Lindell†

March 5, 2010

Abstract

The problem of carrying out cryptographic computations when the participating parties are *rational* in a game-theoretic sense has recently gained much attention. One problem that has been studied considerably is that of rational secret sharing. In this setting, the aim is to construct a mechanism (protocol) so that parties behaving rationally have incentive to cooperate and provide their shares in the reconstruction phase, even if each party prefers to be the only one to learn the secret.

Although this question was only recently asked by Halpern and Teague (STOC 2004), a number of works with beautiful ideas have been presented to solve this problem. However, they all have the property that the protocols constructed need to know the actual utility values of the parties (or at least a bound on them). This assumption is very problematic because the utilities of parties are not public knowledge. We ask whether this *dependence on the actual utility values* is really necessary and prove that in the case of two parties, rational secret sharing cannot be achieved without it. On the positive side, we show that in the multiparty case it is possible to construct a single mechanism that works for all (polynomial) utility functions. Our protocol has an expected number of rounds that is constant, and is optimally resilient to coalitions.

In addition to the above, we observe that the known protocols for rational secret sharing that do not assume simultaneous channels all suffer from the problem that one of the parties can cause the others to output an incorrect value. (This problem arises when a party gains higher utility by having another output an incorrect value than by learning the secret itself; we argue that such a scenario needs to be considered.) We show that this problem is inherent in the non-simultaneous channels model, unless the actual values of the parties' utilities from this attack is known, in which case it is possible to prevent this from happening.

Keywords: rational secret sharing, game theory and cryptography

*This research was supported by THE ISRAEL SCIENCE FOUNDATION (grant No. 781/07). An extended abstract of this work appeared in CRYPTO 2009.

†Department of Computer Science, Bar-Ilan University, ISRAEL. email: gilad_asharov@yahoo.com, lindell@cs.biu.ac.il.

Contents

1 Introduction

1.1 Background

Recently, there has been much interest in the intersection between cryptography and game theory [7, 6, 12, 5, 1, 11, 12]. One specific question that has gained much attention is that of *rational secret sharing*. The basic problem that arises when considering secret sharing (or to be more exact, protocols for the reconstruction phase) is that the parties actually have no incentive to reveal their share. Specifically, assume that t parties get together to reconstruct a secret that was shared using a t-out-of-n secret sharing scheme. The standard way that this reconstruction takes place is simply for each party to broadcast its share to all others. However, if one party does not broadcast its share, it can still reconstruct the secret (because it received the $t-1$ shares of all other parties and so has t shares overall), but the others cannot (because they only have $t-1$ shares). Thus, under the assumption that parties prefer to be the only one to learn the secret, the rational behavior in the above naive reconstruction procedure is for every party to remain quiet and not broadcast its share [7]. The aim of rational secret sharing is therefore to construct a mechanism so that it is in the interest of rational parties to cooperate, with the result being that all parties learn the reconstructed secret. The fact that the parties are rational means that they each have a utility function assigning a value to every possible outcome of the protocol (this value represents the gain that the party achieves if the given outcome occurs). Furthermore, the parties' aim is to maximize their utility. We remark that a mechanism is considered successful if it achieves a Nash equilibrium (or one of its variants) for the strategy which instructs all parties to cooperate. Loosely speaking, this means that if any one of the parties deviates from the prescribed strategy (while others follow it), then it will not obtain a higher utility (and may even lose). Thus, it is in the interest of all parties to follow the prescribed strategy and cooperate.

In order to construct a mechanism with the above properties, certain natural assumptions are made regarding the utilities of the parties. In particular, it is assumed that a party always prefers learning the secret over not learning it (this is essential to assume, or else there is no reason for a party to ever participate in the reconstruction). Furthermore, it is assumed that parties prefer to learn the secret, and have some or all of the other parties not learn it (when knowledge is power, this makes a lot of sense). Although the above assumptions are very reasonable, a concern with all of the known protocols is that they don't just assume that this "learning preference" holds. Rather, they assume that the *actual utility values* of the parties (or at least bounds on them) are known to all, and the mechanism itself depends on these values. The problem with this assumption is that in reality the utility of a party may not even be known to itself, let alone to others. Furthermore, even if a party knows its own utility, it is unclear how others can learn this value (it would not necessarily be rational for a party to be honest about its utility; rather, it may gain something by providing incorrect information about its utility function). This problem stands at the center of this work, and we ask the following fundamental question:

> *Is it possible to construct a single reconstruction mechanism for rational secret sharing that achieves a Nash equilibrium for all possible values of utility functions that fulfill the aforementioned assumptions regarding learning preference?*

In addition to the above, we observe that some of the known protocols suffer from a correctness issue. Specifically, most of the positive results on this topic assumed that the parties have access to a simultaneous channel (meaning that all parties can simultaneously send messages and so no party

can see what the others broadcast before sending its own message). Since simultaneous channels are problematic to implement in practice, a recent breakthrough was made that achieved rational secret sharing in a model with non-simultaneous channels [12]. However, the protocol of [12] (and a follow-up protocol by [8]) has the problem that one of the parties can cause the others to output an incorrect value, at the expense of not learning the secret itself. Thus, the assumption made by [12] is that since a party always prefers to learn the secret, it will never follow such a strategy. However, we do not believe that this assumption is always reasonable. Rather, there are certainly scenarios where a party can gain more by having another learn incorrect information than by learning the information itself (for example, consider the case where the use of incorrect information can result in a loss of reputation, to the potential gain of others). In any case, it would certainly be preferable to not have to assume this. Noting that this problem of correctness does not arise in any of the protocols using simultaneous channels, we ask:

> *Is it possible to construct a reconstruction mechanism for rational secret sharing that uses non-simultaneous channels and achieves Nash equilibrium even if a party's utility when another party outputs an incorrect value is higher than its utility when it learns the secret? Furthermore, is it possible to achieve this without assuming knowledge of the actual utility value?*

1.2 Our Results

We focus mainly on 2-out-of-2 secret sharing. Let U_i^+ denote the utility of party P_i when it learns the secret and the other party does not. Furthermore, let U_i^f denote the utility of party P_i when the other party outputs an incorrect (false) value, even if P_i itself did not learn the output. We call a mechanism U^+-independent if it achieves Nash equilibrium for all possible (polynomial) values of (U_1^+, U_2^+) that fulfill the aforementioned learning-preference assumptions (i.e., that a party prefers learning over not learning, and prefers to be the only one to learn). We define U^f-independence similarly. We stress that when a mechanism is U^+ or U^f-independent, it may still know the values of the other utilities (i.e., the utility when all parties learn the secret or when none learn it). We begin by proving an interesting connection between U^+-independence and *complete fairness*, and between U^f-independence and *correctness* (where fairness and correctness here are in the presence of malicious adversarial behavior that may not be rational and is aimed only to break the protocol). In Section 3, we prove the following informally stated theorem:

Theorem 1.1 *Any two-party mechanism that achieves U^+-independence guarantees complete fairness in the presence of malicious adversarial behavior. Furthermore, any two-party mechanism that achieves U^f-independence guarantees correctness in the presence of malicious adversarial behavior.*

Intuitively, Theorem 1.1 holds because if a mechanism is U^+-independent, then it must be in a party's interest to cooperate even if its U^+ utility is very high. However, if a party's U^+ utility is high enough – but still polynomial – then it can be shown that its best strategy is to just try and break fairness (because then it gains U^+). Since, it should not be able to succeed in doing this, it follows that a malicious adversary also can only break fairness with negligible probability. The connection between U^f independence and correctness is proven in a similar way. It is possible to use Theorem 1.1 in order to prove that there do not exist two-party reconstruction mechanisms for rational secret sharing that are independent of U^+, by showing how to toss a fair coin given

any such mechanism. (Intuitively, given such a mechanism, we construct a protocol where in the first stage multiparty computation is used to generate shares of an unbiased coin, and then the mechanism is used to fairly reveal the coin.) Using the impossibility result of Cleve [4] for coin tossing, we then conclude that such a mechanism does not exist. However, we stress that unbiased coin tossing is only impossible in the non-simultaneous channels model, and thus this would only prove the impossibility of obtaining U^+-independence in this model, and leaves open the possibility that there do exist U^+-independent mechanisms in the simultaneous channels model.

We therefore provide a direct proof, ruling out the possibility of obtaining U^+-independence even when given a simultaneous channel. That is, we prove the following:

Theorem 1.2 *There does not exist a two-party reconstruction mechanism for rational secret sharing that is independent of U^+ in either the simultaneous or non-simultaneous channels model.*

In order to prove this, we present a lower bound on the number of rounds needed for achieving fair reconstruction and show that this number is dependent on the actual utility functions of the parties (or, to be more exact, a bound on them). Thus, no mechanism can be independent of the utilities because this implies that its number of rounds is also independent. Our lower bound is proven in the simultaneous-channels model and therefore also holds for non-simultaneous channels.

Having established that U^+-independence is impossible to achieve, we ask whether the other utility values must also be known. For example, we know that U^f-independence is possible in the simultaneous-channels model, because all of the known protocols for the simultaneous-channels model (cf. [6, 12]) are U^f-independent. This leaves open the question regarding U^f-independence with non-simultaneous channels. We prove that:

Theorem 1.3 *There does not exist a two-party reconstruction mechanism for rational secret sharing that is U^f-independent in the non-simultaneous channels model.*

The proof of this theorem uses Theorem 1.1 that states that a U^f-independent mechanism guarantees correctness. We then prove that in the non-simultaneous channels model, a *correct* reconstruction mechanism cannot be *fair*.

Positive results. In Section 5, we present two positive results as follows:

1. We present a *multiparty* reconstruction mechanism that uses simultaneous channels and is *independent of all utility values*. The reconstruction mechanism is also resilient to coalitions of size $t/2$, where t is the secret sharing threshold (i.e., when considering t-out-of-n secret sharing). We also show that it is not possible to achieve resilience to coalitions of a larger size, while preserving utility independence. Thus, in this sense, our mechanism is optimal.[1]

2. We present a two-party reconstruction mechanism for rational secret sharing that works in the non-simultaneous model and achieves correctness. This mechanism uses the actual values of U^f (recall that by Theorem 1.3 U^f independence is impossible and so the use of the actual values of U^f here is inherent).

The above results show that **(a)** utility independence is possible to achieve in the multiparty setting, and **(b)** correctness need not be forfeited in the model with non-simultaneous channels.

[1] We note that in the preliminary version of this paper that appeared at CRYPTO 2009, we presented a mechanism that achieved utility independence when assuming a *relaxation* on the assumptions of the utilities functions of the parties. Here, we present a stronger result which does not need any relaxation at all.

1.3 Related work.

The question of rational secret sharing was first introduced by [7]. They showed that there does not exist a mechanism with a constant number of rounds, that achieves Nash Equilibrium that survives iterated deletions of weakly dominated strategies. Moreover, they presented a protocol for $n \geq 3$ (that is U^+-dependent) in the simultaneous model. More protocols, dealing with other settings, were presented for the simultaneous model in [6, 1, 11, 12], and for the non-simultaneous model in [12, 8]. The basic question that we ask regarding utility independence was proposed in [7]. The first partial answer to this question was given by [1] (with extension in [2]) who showed that utility independence is possible for t-out-of-n secret sharing as long as $t < n/3$. This question was also considered by [16] who showed that utility independence can be achieved if the number of parties participating in the reconstruction procedure is strictly greater than the threshold t. The works of [15, 13] can be used to obtain fair secret sharing, but assume stronger physical assumptions than a simultaneous channel. Other works have also considered a mix of rational, honest and malicious parties [18, 16, 1].

2 Preliminaries and Definitions

2.1 Secret Sharing

Informally, a t out of n secret sharing scheme involves a dealer D and n parties $P_1, .., P_n$. The dealer D wishes to distribute "shares" of a secret s which is chosen according to some efficiently samplable distribution ensemble $\mathcal{S} = \{\mathcal{S}_k\}_{k\in\mathbb{N}}$. The main security requirement is that every subset of $t^* \geq t$ parties is capable of reconstructing s, whereas any subset of $t^* < t$ parties cannot learn anything whatsoever about the secret. See [17] for a formal definition.

A secret sharing protocol consists of two phases, called sharing and reconstruction. In the first phase the dealer selects a secret s according the distribution $\mathcal{S}_k$, where k is the security parameter. It then generates the shares from the secret, and sends each party its share using a private channel (we assume that the dealer can distribute the shares perfectly privately). The algorithm used to generate the shares for each party is denoted $\text{SHARE}(\mathcal{S})$, and the shares themselves are denoted $s_1, \ldots, s_n$. We remark that the above assumes that $\text{SHARE}(\mathcal{S})$ is a non-interactive procedure. In the second phase, the parties run an interactive protocol in order to reconstruct the secret. We stress that the dealer is not involved in this phase, and we assume for simplicity that the participating parties can communicate via a broadcast channel. At the end of the reconstruction protocol, each party outputs what it has "learned" (i.e., what it believes that the share is).

The broadcast channel is sometimes *simultaneous*, meaning that some parties can broadcast messages at the same time (and so the adversary is not rushing), and sometimes the channel is *not simultaneous*, and so there is only a single sender at any one time (or, equivalently, the adversary is assumed to be rushing). Clearly, it is preferable to have protocols for the non-simultaneous model as true simultaneity is very hard (if not impossible) to achieve in practice.

2.2 Game Theory and Fair Secret Sharing

We assume that the parties are *rational*. Therefore, we will refer to the reconstruction protocol as a *mechanism*, where a mechanism is a pair $(\Gamma, \vec{\sigma})$, where Γ is the game (i.e., a specification of what actions are allowed or possible by the parties), and $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ is a strategy for that game (σ_i

is a set of instructions for party P_i in the game). The aim of a mechanism is to have the parties follow the prescribed strategy $\vec{\sigma}$, which in turn will result in the desired outcome. Another way of looking at this is that the recommended strategy $\vec{\sigma}$ is a protocol, and the game Γ is defined to be all possible deviations from the protocol.

Notation: We call a vector of parties' strategies a *strategy profile*, and use the following notations: $\vec{\alpha}_{-i} = (\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_n)$, $(\alpha'_i, \vec{\alpha}_{-i}) = (\alpha_1, ..., \alpha_{i-1}, \alpha'_i, \alpha_{i+1}, ..., \alpha_n)$, and $u_i(\sigma'_i, \vec{\sigma}_{-i})$ is the expected utility of party i when it plays according to σ'_i, while other parties play according to $\vec{\sigma}$. When considering 2-out-of-2 secret sharing, we sometimes use the notation $u_i(\sigma_1, \sigma'_2)$, which is the expected utility of party $i \in \{1, 2\}$, when party P_1 plays according to σ_1, and P_2 plays according to σ'_2. Again, in the two party setting, letting $i \in \{1, 2\}$ we denote by P_{-i} the party P_{3-i} (i.e., the participating parties are P_i and P_{-i}).

Definition 2.1 (Nash Equilibrium:) *A behavioral strategy profile $\vec{\sigma}$ for the game Γ is said to be a* Nash equilibrium *if for every $i \in [n]$ and every behavioral strategy σ'_i for P_i, it holds that $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$.*

Definition 2.2 (ϵ-Nash Equilibrium:) *Let $\epsilon : \mathbb{N} \to [0, 1]$ be a function. A behavioral strategy profile $\vec{\sigma}$ for the game Γ is said to be an ϵ-*Nash equilibrium *if for every $i \in [n]$ and every behavioral strategy σ'_i, it holds that $u_i(\sigma'_i, \vec{\sigma}_{-i}) \leq u_i(\sigma_i, \vec{\sigma}_{-i}) + \epsilon(k)$.*

Definition 2.3 (Strict Nash Equilibrium:) *A behavioral strategy profile $\vec{\sigma}$ for the game Γ is said to be a* strict Nash equilibrium *if for every $i \in [n]$ and every behavioral strategy $\sigma'_i \neq \sigma_i$, it holds that $u_i(\sigma_i, \vec{\sigma}_{-i}) > u_i(\sigma'_i, \vec{\sigma}_{-i})$.*

Computational Nash Equilibrium. In a computational world, all parties run in probabilistic polynomial-time, and events that happen with negligible probability are not of any concern. We therefore consider probabilistic polynomial-time behavioral strategies and ϵ-Nash for a negligible function ϵ. Nash equilibrium in such a world was defined by [5, 9, 8], as follows:

Definition 2.4 (Computational Nash Equilibrium:) *A behavioral strategy profile $\vec{\sigma}$ for the game Γ is said to be a* Computational Nash equilibrium *if it is probabilistic polynomial-time and there exists a negligible function $\epsilon : \mathbb{N} \to [0, 1]$ such that $\vec{\sigma}$ is an $\epsilon(k)$-Nash equilibrium where k denotes the security parameter.*

Outcome and utilities. The outcome of an execution of a game Γ with some strategy profile $\vec{\sigma}$ is denoted o and consists of the output of all of the parties. In the case of 2-out-of-2 secret sharing, each party may learn or may not learn the secret, and there are therefore exactly four possible outcomes. (This ignores the issue of correctness which we introduce in this work and discuss below.) Each party's utility is a function of these outcomes, and there are therefore also four possible utility values for each party. The notations for the four possible outcomes, and the associated utility for each party, are described in Table 1.

In this work, we consider the possibility that parties may output incorrect values and introduce a utility U^f for this event (informally, a party gains U_i^f if it succeeds in having the other party output a false/incorrect value). This results in *nine* possible outcomes of the game (each party may learn the correct value, not learn, or output an incorrect value). For simplicity we will consider

P_1 receives s	P_2 receives s	Outcome notation	P_1's Utility	P_2's Utility
NO	NO	o^{none}	U_1^-	U_2^-
NO	YES	o_2^+	U_1^{--}	U_2^+
YES	NO	o_1^+	U_1^+	U_2^{--}
YES	YES	o^{both}	U_1	U_2

Table 1: Outcome and Utility

only the outcome where one party does not learn the secret while the other outputs an incorrect (or false) value. We denote this event by o_{-i}^{false} where P_{-i} is the party who outputs the incorrect value. (We explicitly consider this event because this is the one that occurs naturally. Needless to say, when analyzing mechanisms all possibilities need to be taken into account.)

Assumptions on the utility functions. We assume that the utility functions of all parties are polynomial in the security parameter. The importance of this is discussed in Appendix A. Formally, a party's utility function u_i is a function of the outcome and the security parameter k. We therefore write $U_i(1^k) = u_i(1^k, o^{\text{both}})$, $U_i^+(1^k) = u_i(1^k, o_i^+)$, $U_i^-(1^k) = u_i(1^k, o^{\text{none}})$, $U_i^{--}(1^k) = u_i(1^k, o_{-i}^+)$, and $U_i^f(1^k) = u_i(1^k, o_{-i}^{\text{false}})$. As is now standard [7, 6, 12], we assume that each party always prefers learning the secret over not learning it, and that each party most prefers to be the sole party to learn the secret. We add an additional assumption being that a party prefers to have the other party output an incorrect value than not, when in both cases the first party does not learn anyway. We do not make any assumption on U_i^f beyond this. (In [12] they implicitly assume that $U_i^f < U_i$ for all parties.) For lack of a better name, we call utility functions that fulfill these assumptions "natural". Formally:

Definition 2.5 *Let* $\mathcal{U} = \left\{ \left(U_i^+, U_i, U_i^-, U_i^{--}, U_i^f \right)_{i \in \{1,2\}} \right\}$ *be a set of utility functions for the parties. We say that* $\mathcal{U}$ *is* natural *if for every* $i \in \{1,2\}$ *and for every* $k \in \mathbb{N}$ *it holds that*

$$U_i^+(k) \geq U_i(k) \geq U_i^-(k) \geq U_i^{--}(k) \geq 0 \qquad \text{and} \qquad U_i^f(k) \geq U_i^-(k).$$

We remark that in some previous works, it was formally assumed that $U_i^-(k) = U_i^{--}(k)$, even though none of the protocols utilized this fact (this is despite the fact that [7] explicitly introduce this utility value). We find it unsatisfactory to assume that once a party has not learned, it makes no difference to its utility if others did or did not learn. On the contrary, it can be a lot worse if a party does not learn while others do learn and so protocols should take this into account.

Fair secret sharing. A number of different notions have been used regarding the desired equilibrium for rational secret sharing. Our impossibility results refer to the weakest of these assumptions, which is ϵ-Nash equilibrium for a negligible function $\epsilon(\cdot)$ [12, 9]. However, we also require that the number of rounds be polynomial (this is needed for our lower bounds, but we argue that this does not significantly weaken our results because a mechanism with a super-polynomial of rounds is not computationally feasible to run). The natural way to model this is as a computational Nash equilibrium [5, 9]. We define computationally fair reconstruction mechanisms in this light:

Definition 2.6 *Let* $\mathcal{U}$ *be a set of natural utility functions for* P_1 *and* P_2 *(as in Definition 2.5). We say that a mechanism* $(\Gamma, \vec{\sigma})$ *is a* fair reconstruction mechanism for $\mathcal{U}$ *if* $\vec{\sigma}$ *is a computational Nash*

Equilibrium and if the probability that the result is not o^{both} *when both parties follow* $\vec{\sigma}$ *is negligible.*

3 Utility-Independent Mechanisms and Properties

3.1 Definitions

We now formalize the notion of utility independence. Loosely speaking, a mechanism is independent of a given utility function if it achieves its desired properties for *any* value of that utility for all parties.

Definition 3.1 (utility independence): *Let* $\hat{U} \in \{U^+, U, U^-, U^{--}, U^f\}$ *be a utility type and let* $\mathcal{U}' = \{U_i^+, U_i, U_i^-, U_i^{--}, U_i^f\}_{i=1}^n \setminus \{\hat{U}_i\}_{i=1}^n$ *be a set of polynomial utility functions (excluding all the* $\hat{U}_i$ *values). We say that the mechanism* $(\Gamma, \vec{\sigma})$ *is a* $\hat{U}$-independent fair reconstruction mechanism *if for all polynomial utility functions* $\{\hat{U}_i\}_{i=1}^n$ *for which* $\mathcal{U} = \mathcal{U}' \cup \{\hat{U}_i\}_{i=1}^n$ *is natural, it holds that* $(\Gamma, \vec{\sigma})$ *is a fair reconstruction mechanism for* $\mathcal{U}$*.*

Note that our definition of utility independence includes the assumption that $\mathcal{U}$ is natural. In our results, we focus on U^+ and U^f-independence.

Fairness and correctness. In this section, we show that U^+ and U^f-independence, respectively, imply the properties of complete fairness and correctness in the presence of adversarial behavior. We stress that we define these notions in an *adversarial context* and not in a game theoretic one. That is, we say that a protocol or mechanism is completely fair/correct if it maintains this property when one of the parties follows a worst-case strategy (meaning that it has no aim to gain utility and its aim is simply to break this property of the protocol). Before proceeding, we remark that we will freely move between protocols in a cryptographic setting with an adversary $\mathcal{A}$ and mechanisms involving rational adversaries playing a game in order to achieve utility. Despite the apparent differences between these notions, they are actually very similar. In particular, one can define a one-to-one mapping ϕ from a mechanism to a protocol and back, as follows:

- Let $(\Gamma, \vec{\sigma})$ be a mechanism. We define $\pi = \phi(\Gamma, \vec{\sigma})$ to be a protocol where all honest parties follow the strategy specified in $\vec{\sigma}$, according to the game Γ.
- Let π be a protocol. We define $(\Gamma, \vec{\sigma}) = \phi^{-1}(\pi)$ to be the mechanism where Γ specifies the order of sending messages that appears in π, and $\vec{\sigma}$ is the strategy to follow the instructions of π honestly.

We now proceed to define complete fairness and correctness. We present the definitions in a "protocol context"; their translation to the game-theoretic context is discussed below. Intuitively, a two-party reconstruction protocol is *completely fair* if whenever one party learns the secret the other party is also guaranteed to learn the secret, except with negligible probability. Likewise, a reconstruction protocol is *correct* if the honest party is guaranteed to either output the correct value (i.e., the secret that was shared) or a special abort symbol $\perp$. Although it is difficult to formalize these notions for general secure computation without resorting to a full ideal model/real model definition (since the output depends on the actual inputs used by the possibly malicious parties), in the case of secret sharing it is much simpler because the output of the protocol is well defined. In particular, the output can only be the shared secret s or an abort symbol $\perp$. We assume that

any reconstruction protocol is non-trivial meaning that if both parties are honest, then they both learn the secret except with negligible probability.

In order to formalize the above, we introduce some notation. Let $\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))$ denote the outcome o of an execution of the reconstruction protocol π, with the parties P_1 and P_2, an adversary $\mathcal{A}$ controlling party P_i ($i \in \{1,2\}$), and a share s that was chosen according to the distribution $\mathcal{S}$ and shared as in SHARE; recall that an outcome is simply the concatenation of the outputs of all participating parties (since $\mathcal{A}$ controls P_i, we consider only the output of $\mathcal{A}$ and the honest party). Next, denote by $\text{OUTPUT}_X(\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))$ the output of party X (where X may be $\mathcal{A}$ or the honest party P_{-i}). Recall that the security parameter is denoted k.

Definition 3.2 *Let* SHARE *be a share generation algorithm for a 2-out-of-2 secret sharing scheme, and let π be the reconstruction protocol for the scheme.*

1. *We say that π is* completely fair *if for every probabilistic polynomial-time adversary $\mathcal{A}$ that controls party P_i there exists a negligible function $\mu(\cdot)$ such that*

$$\Pr[\text{OUTPUT}_{\mathcal{A}}(\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))) = \mathcal{S}] \leq \Pr[\text{OUTPUT}_{P_{-i}}(\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))) = \mathcal{S}] + \mu(k)$$

2. *We say that π is* correct *if for every probabilistic polynomial-time adversary $\mathcal{A}$ that controls party P_i there exists a negligible function $\mu(\cdot)$ such that*

$$\Pr[\text{OUTPUT}_{P_{-i}}(\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))) \notin \{\mathcal{S}, \perp\}] \leq \mu(k).$$

An equivalent formulation of the above for mechanisms is obtained by requiring that the result of an execution where one party follows the prescribed strategy and the other may follow *any arbitrary alternative* strategy is fair (or correct). For example, correctness of a mechanism $(\Gamma, \vec{\sigma})$ can be formalized by saying that for *every arbitrary strategy* σ_i' followed by party P_i ($i \in \{1,2\}$) there exists a negligible function μ such that:

$$\Pr[\text{OUTPUT}_{P_{-i}}(\text{REAL}_{\Gamma,P_i(\sigma_i'),P_{-i}(\sigma_{-i})}(\text{SHARE}(\mathcal{S}))) \notin \{\mathcal{S}, \perp\}] \leq \mu(k).$$

(Observe that correctness is guaranteed only when party P_{-i} follows the prescribed strategy σ_{-i}.)

3.2 U^+-Independence Implies Fairness

We now prove that the existence of a U^+-independent reconstruction mechanism implies the existence of a completely fair reconstruction protocol. Intuitively this holds because if complete fairness is not achieved, then there exists an adversary who can participate in the protocol induced from the mechanism and with non-negligible probability can learn the secret while the honest party does not. Given such an adversary, we can set the utility U^+ of one of the parties to be high enough so that its expected gain by following the adversarial strategy is high enough. Our proof holds for both simultaneous and non-simultaneous channels.

Proposition 3.3 *If there exists a U^+-independent fair reconstruction mechanism for a 2-out-of-2 secret sharing scheme (as in Definition 3.1), then there exists a completely fair reconstruction protocol (as in Definition 3.2) for the scheme.*

Proof: Let $(\Gamma, \vec{\sigma})$ be a U^+-independent fair reconstruction mechanism and let $\mathcal{U}'$ be a set of utilities specifying $\{U, U^-, U^{--}, U^f\}$ for both parties. Denote by π the protocol derived from $(\Gamma, \vec{\sigma})$ by the mapping ϕ described above in Section 3.1. Assume by contradiction that π is not a completely fair reconstruction protocol. This implies that there exists a probabilistic polynomial-time adversary $\mathcal{A}$ that controls some party P_i ($i \in \{1,2\}$) and a polynomial $p(\cdot)$ such that for infinitely many k's:

$$\Pr\left[\text{OUTPUT}_{\mathcal{A}}\left(\text{REAL}_{\pi,\mathcal{A},i}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right) = \mathcal{S}\right] > \Pr\left[\text{OUTPUT}_{P_{-i}}\left(\text{REAL}_{\pi,\mathcal{A},i}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right) = \mathcal{S}\right] + \frac{1}{p(k)}$$

Let $\sigma^{\mathcal{A}}$ be the corresponding behavioral strategy of the adversary $\mathcal{A}$ in the game Γ. Note that for infinitely many k's, the outcome of the game when party P_i plays according to $\sigma^{\mathcal{A}}$, while the other party plays according to the prescribed strategy $\vec{\sigma}$, is o_i^+ with probability $1/p(k)$.

We now define the utility function U_i^+ for party P_i by $U_i^+ \geq p(k) \cdot (U_i + 1)$. We show that for infinitely many k's, P_i's utility is greater if it follows $\sigma^{\mathcal{A}}$ than if it follows σ_i, which is a contradiction to the assumption that $\vec{\sigma}$ is a (computational) Nash equilibrium. Let $\mathcal{O}$ denote the set of all possible outcomes, and recall that $u_i(o)$ is the utility of P_i upon outcome o. We have that for infinitely many k's:

$$\begin{aligned} u_i\left(\sigma_i^{\mathcal{A}}, \sigma_{-i}\right) &= \sum_{o \in \mathcal{O}} \Pr[o \mid (\sigma_i^{\mathcal{A}}, \sigma_{-i})] \cdot u_i(o) \\ &\geq \Pr\left[o_i^+ \mid (\sigma_i^{\mathcal{A}}, \sigma_{-i})\right] \cdot U_i^+ \\ &\geq \frac{1}{p(k)} \cdot (p(k) \cdot (U_i + 1)) = U_i + 1. \end{aligned}$$

In contrast,

$$u_i\left(\sigma_i, \sigma_{-i}\right) = U_i.$$

Thus, there exists a non negligible function ϵ' (even if U_i is negligible), such that:

$$u_i\left(\sigma_i^{\mathcal{A}}, \sigma_{-i}\right) \geq u_i\left(\sigma_i, \sigma_{-i}\right) + \epsilon'(k)$$

in contradiction to the assumption that $\vec{\sigma}$ is a computational Nash equilibrium for Γ. We therefore conclude that the protocol π induced from $(\Gamma, \vec{\sigma})$ is completely fair, as in Definition 3.2. ■

3.3 U^f-Independence Implies Correctness

In this section, we prove that U^f independence implies correctness. That is, we show that any protocol that is U^f-independent achieves correctness in the presence of malicious adversarial behavior. The intuition for this proof is the same as for U^+ and fairness. Namely, if the mechanism does not achieve correctness then the strategy used to break it can provide a high enough payoff, given a large enough U^f.

Proposition 3.4 *If a fair reconstruction mechanism for a 2-out-of-2 secret sharing scheme is U^f-independent (as in Definition 3.1), then it achieves correctness (as in Definition 3.2).*

Proof: Let $(\Gamma, \vec{\sigma})$ be a U^f-independent mechanism, and assume by contradiction that the mechanism does not imply correctness as in Definition 3.2. We will show that for some $i \in \{1,2\}$ and a particular value of U_i^f, the prescribed strategy $\vec{\sigma}$ is not a computational Nash Equilibrium. By

our contradicting assumption, there exists an adversary $\mathcal{A}$ that controls party P_i and a polynomial $p(\cdot)$ such that for infinitely many k's:

$$\Pr[\text{OUTPUT}_{P_{-i}}(\text{REAL}_{\pi,\mathcal{A},i}(\text{SHARE}(\mathcal{S}))) \notin \{\mathcal{S}, \perp\}] \geq \frac{1}{p(k)}$$

Let $\sigma^{\mathcal{A}}$ be the corresponding behavioral strategy of $\mathcal{A}$ in the game Γ. Let $\mathcal{O}$ denote the set of all possible outcomes, and recall that $u_i(o)$ is the utility of P_i upon outcome o. It follows that for infinitely many k's, the expected utility of party P_i when it plays according to $\sigma^{\mathcal{A}}$, and assuming that P_{-i} plays according to σ_i, is:

$$\begin{aligned} u_i(\sigma^{\mathcal{A}}, \sigma_{-i}) &= \sum_{o \in \mathcal{O}} \Pr[o \mid (\sigma_i^{\mathcal{A}}, \sigma_{-i})] \cdot u_i(o) \geq \Pr\left[o_{-i}^{false} \mid (\sigma_i^{\mathcal{A}}, \sigma_{-i})\right] \cdot u_i(o_{-i}^{false}) \\ &\geq \frac{1}{p(k)} \cdot U_i^f \end{aligned}$$

Now, if $U_i^f > p(k) \cdot (U_i + 1)$, then we have that:

$$u_i(\sigma^{\mathcal{A}}, \sigma_{-i}) \geq \frac{1}{p(k)} \cdot U_i^f > \frac{1}{p(k)} \cdot p(k) \cdot (U_i + 1) = U_i + 1$$

It follows that $u_i(\sigma^{\mathcal{A}}, \sigma_{-i}) > U_i + 1$ and so $\vec{\sigma}$ is not an ϵ-Nash Equilibrium for any negligible ϵ; in particular $\vec{\sigma}$ is not a computational Nash equilibrium. ■

4 Negative Results

4.1 Impossibility for U^+-Independence

As we have mentioned, Proposition 3.3 can be used to prove the impossibility of obtaining U^+-independent fair reconstruction mechanisms in the non-simultaneous channels model. This is because any such mechanism can be used to toss a fair coin, in contradiction to [4]. (Specifically, secure computation can be used to generate shares of a random bit, which are then reconstructed using the mechanism. By Proposition 3.3, this mechanism guarantees complete fairness in the presence of malicious behavior and so neither party can bias the outcome.) Such a proof leaves open the possibility of obtaining U^+-independence in the simultaneous channels model. In this section we therefore prove a lower bound on the number of rounds that are needed in any fair reconstruction mechanism, even in the simultaneous model. As we will see, the number of rounds depends on the U^+ utilities of the parties; U^+-independence is therefore not achievable.

Completeness assumption. We assume that every execution of the protocol, where *both* parties are honest, will end at a point where *both* parties learn the secret, except with some negligible probability. Let $(\Gamma, \vec{\sigma})$ be the mechanism, where Γ is the game and $\vec{\sigma}$ is the prescribed strategy. The completeness assumptions states that:

$$\begin{aligned} \Pr\Big[&\text{OUTPUT}_{P_1}\left(\text{REAL}_{\Gamma,P_1(\sigma),P_2(\sigma)}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right) \\ &= \text{OUTPUT}_{P_2}\left(\text{REAL}_{\Gamma,P_1(\sigma),P_2(\sigma)}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right) = \mathcal{S}\Big] \geq 1 - \mu(k) \end{aligned}$$

for some negligible function μ.

Number of rounds as random variable. Protocols for rational reconstruction all have an expected number of rounds (this was proven to be necessary by [7]), and so the number of rounds of a protocol is actually a random variable. We denote by $\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma),P_2(\sigma)}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right)$ the number of rounds of the execution of the game Γ with parties P_1 and P_2 where both parties follow the prescribed strategy $\vec{\sigma}$ and use the shares generated by the algorithm SHARE. For brevity, we denote it by $R^{\Gamma}_{(\sigma_1,\sigma_2)}$.

Additional assumption on the utility functions of the parties: We add the assumption that there exists a non-negligible difference between U (the value that the party gains when both parties learn the secret) and U_i^{--} (the value that the party gains when it does not learn the secret, but the other party does). That is, there exists a non-negligible function ϵ' such that for every i and k,

$$U_i(k) - U_i^{--}(k) > \epsilon'(k). \tag{1}$$

This assumption makes a lot of sense. In particular, assuming that any acceptable mechanism does not enable party P_i to obtain U_i^{+}, the party may as well just hand its share to the other party and gain utility U_i^{--} (which is the same as U_i up to a negligible amount).

Intuition for the Theorem: We prove our lower bound by considering a specific attack (or, an alternative strategy) that can be carried out on every mechanism. The attack that we consider is a premature abort. When a party aborts prematurely, it does not broadcast its message in the round that it quits, while the other party does. Therefore, intuitively, it may gain more information about the secret than the other party. The mechanism must therefore guarantee that the amount of information gained in any single round is small enough so that carrying out such an attack is not profitable and will yield a lower utility. We quantify this amount of information and define an "aborting threshold" for each party as follows:

$$\beta_1 = \frac{U_1 - U_1^{--}}{U_1^{+} - U_1^{--}} \qquad \text{and} \qquad \beta_2 = \frac{U_2 - U_2^{--}}{U_2^{+} - U_2^{--}}.$$

Notation: Denote by a_i the output of party P_1 when P_2 quits at round i *before* sending its message (that is, at round i only P_1 broadcast its message); likewise b_i denotes the output of P_2 when P_1 quits at round i. We call these values "the default outputs"; see Figure 1.

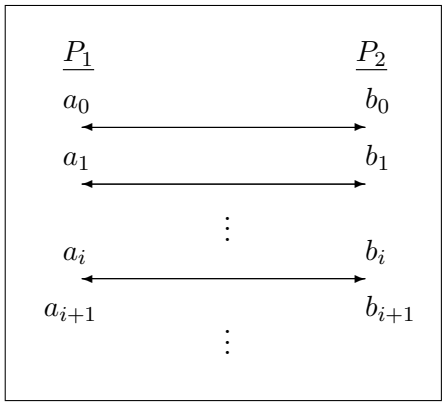


Figure 1: Default output notations (simultaneous)

Note that when P_1 quits at round i (before sending its message) and P_2 does not quit in that round, party P_1 receives an additional message and therefore may gain additional knowledge about

the secret. In such a case, P_1 outputs a_{i+1}, while P_2 outputs b_i. In the following claim, we bound the amount of additional knowledge that a party can gain in such a situation:

Claim 4.1 *Let $\mathcal{U}$ be a set of natural utility functions for P_1 and P_2 (as in Definition 2.5), and let the mechanism $(\Gamma, \vec{\sigma})$ be a* fair reconstruction mechanism for $\mathcal{U}$ *(as in Definition 2.6). For every round $i \geq 0$, the following must hold:*

1. $\Pr[a_{i+1} = s] \leq \Pr[b_i = s] + 2\beta_1$
2. $\Pr[b_{i+1} = s] \leq \Pr[a_i = s] + 2\beta_2$

Proof: We show that the first requirement must hold; the second can be shown in similar way. Assume by contradiction that there exists a round i such that:

$$\Pr[a_{i+1} = s] > \Pr[b_i = s] + 2\beta_1$$

We show that $\vec{\sigma}$ is not a computational-Nash Equilibrium in Γ. Denote by σ_1^i the strategy when party P_1 acts according to the prescribed strategy σ_1 in rounds 1 through i, and does not broadcast its message at round $i+1$. We show that it is better for P_1 to play according to σ_1^i then σ_1, when P_2 plays according to the prescribed strategy σ_2. In order to show this, we need to show that the expected utility of P_1 when it plays according to σ_i^1 is greater than the expected utility when it plays according to σ_1 (when in both cases we assume that P_2 plays according to σ_2). That is, we need to show that $u_1(\sigma_1^i, \sigma_2) > u_1(\sigma_1, \sigma_2) + \epsilon'$, for some non-negligible function ϵ'. Using the fact that $(\Gamma, \vec{\sigma})$ is fair, we know that $u_1(\sigma_1, \sigma_2) = U_1$. We need to calculate:

$$\begin{aligned} u_1\left(\sigma_1^i, \sigma_2\right) &= \sum_{o \in \mathcal{O}} \Pr\left[o \mid (\sigma_1^i, \sigma_2)\right] \cdot u_1(o) \\ &\geq \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right] \cdot U_1^+ + \left(1 - \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right]\right) \cdot U_1^{--} \end{aligned} \quad (2)$$

The above holds since U_1^{--} is the lowest utility that P_1 can gain. In order to continue, we need to calculate $\Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right]$. When P_1 plays according to strategy σ_1^i, the only way that it can be the only one to learn the secret is if it learns the secret in round $i+1$ and P_2 has not yet learned the secret. Note that P_1 plays honestly up until round $i+1$ and so if the protocol terminates before round $i+1$ it must be that both parties learned the secret. Therefore, P_1 is the only one to learn the secret if and only if round $i+1$ is reachable (that is, $R^{\Gamma}_{(\sigma_1,\sigma_2)} > i$), and $a_{i+1} = s$ while $b_i \neq s$. We therefore have:

$$\begin{aligned} \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right] &= \Pr\left[a_{i+1} = s \wedge b_i \neq s \wedge R^{\Gamma}_{(\sigma_1,\sigma_2)} > i\right] \\ &= \Pr[a_{i+1} = s \wedge b_i \neq s] \cdot \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} > i \mid a_{i+1} = s, b_i \neq s\right] \end{aligned} \quad (3)$$

Calculating $\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} > i \mid a_{i+1} = s, b_i \neq s\right]$: Using the completeness assumption, the fact that $b_i \neq s$ implies that the number of rounds is greater than i, except with negligible probability. That is:

$$\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} > i \mid a_{i+1} = s, b_i \neq s\right] \geq 1 - \mu(k)$$

where μ is some negligible function.

Calculating $\Pr[a_{i+1} = s \wedge b_i \neq s]$: Using the inequality $\Pr[A \wedge \neg B] \geq \Pr[A] - \Pr[B]$ we have that:

$$\Pr[a_{i+1} = s \wedge b_i \neq s] \quad \geq \quad \Pr[a_{i+1} = s] - \Pr[b_i = s]$$

Recall that by our contradiction assumption $\Pr[a_{i+1} = s] > \Pr[b_i = s] + 2\beta_1$. Therefore:

$$\Pr[a_{i+1} = s \wedge b_i \neq s] > 2\beta_1$$

Completing the proof of Claim 4.1: By Eq. (3), we have:

$$\begin{aligned} \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right] &= \Pr[a_{i+1} = s \wedge b_i \neq s] \cdot \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} > i \mid a_{i+1} = s, b_i \neq s\right] \\ &> 2\beta_1 \cdot (1 - \mu(k)) = 2\beta_1 - \mu'(k) \end{aligned}$$

for some negligible function μ', where the latter is true since β is polynomial and μ is negligible. Plugging this into Eq. (2) we have:

$$\begin{aligned} u_1\left(\sigma_1^i, \sigma_2\right) &\geq \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right] \cdot U_1^+ + \left(1 - \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right]\right) \cdot U_1^{--} \\ &= U_1^{--} + \Pr\left[o_1^+ \mid \left(\sigma_1^i, \sigma_2\right)\right] \cdot \left(U_1^+ - U_1^{--}\right) \\ &> U_1^{--} + (2\beta_1 - \mu') \cdot \left(U_1^+ - U_1^{--}\right) \\ &= U_1^{--} + 2 \cdot \frac{U_1 - U_1^{--}}{U_1^+ - U_1^{--}} \cdot \left(U_1^+ - U_1^{--}\right) - \mu' \cdot \left(U_1^+ - U_1^{--}\right) \\ &= U_1^{--} + 2U_1 - 2U_1^{--} - \mu' \cdot \left(U_1^+ - U_1^{--}\right) \\ &= U_1 + (U_1 - U_1^{--}) - \mu' \cdot \left(U_1^+ - U_1^{--}\right) \end{aligned}$$

Using the assumption that there is a non-negligible difference between U_1 and U_1^{--} (see Eq. (1)), and using the fact that μ' is a negligible function, we conclude that:

$$\begin{aligned} u_1\left(\sigma_1^i, \sigma_2\right) &\geq U_1 + (U_1 - U_1^{--}) - \mu' \cdot \left(U_1^+ - U_1^{--}\right) \\ &= U_1 + \epsilon' \end{aligned}$$

for some non-negligible function ϵ'. This contradicts the assumption that $\vec{\sigma}$ is a computational Nash equilibrium in Γ, completing the proof of Claim 4.1. ■

Claim 4.2 *Let $\beta \leq \min\{\beta_1, \beta_2\}$. For every i it holds that:*

1. $\Pr[a_i = s] \leq 2i\beta + \mu(k)$
2. $\Pr[b_i = s] \leq 2i\beta + \mu(k)$

Where μ is some negligible function.

Proof Sketch: Since the mechanism must work for all samplable distributions $\mathcal{S}$ over the secret, it also has to work for the uniform distribution over $\{0,1\}^k$. Observe that in this case, the probability that the parties output the correct secret without any interaction is 2^{-k}. The claim is proven by induction. The base case follows from the fact that when $i = 0$ the parties can guess the secret with only negligible probability, and the inductive step follows from Claim 4.1. ■

The main theorem: We use the above claims in order to show our main theorem. We now prove that the number of rounds in any fair reconstruction mechanism depends on $\{\beta_1, \beta_2\}$ and so depends on the actual utilities.

Theorem 4.3 *Let $(\Gamma, \vec{\sigma})$ be a fair reconstruction mechanism, let $R^{\Gamma}_{(\sigma_1,\sigma_2)}$ be a random variable denoting the number of rounds in Γ when both parties play according to $\vec{\sigma} = (\sigma_1, \sigma_2)$, and let $\beta \leq \min\{\beta_1, \beta_2\}$ be as above. Then:*

$$E[R^{\Gamma}_{(\sigma_1,\sigma_2)}] > \frac{1}{8\sqrt{\beta}}$$

Proof: We start by calculating an upper bound on the event $R^{\Gamma}_{(\sigma_1,\sigma_2)} = i$. Recall that the protocol may fail with some negligible probability, even when both parties are honest. Let good be the event that both parties learn the secret when both play honestly, and let bad be the event that at least one of the parties does not learn the secret when both parties play honestly. By our completeness assumptions, we know that:

$$\Pr[\mathsf{good}] \geq 1 - \mu(k) \qquad \text{and} \qquad \Pr[\mathsf{bad}] \leq \mu(k).$$

Note that when the number of rounds is i, and we are in good, then both $a_i = s$ and $b_i = s$. Therefore, we have:

$$\begin{aligned}
\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} = i\right] &= \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} = i \mid \mathsf{good}\right] \cdot \Pr[\mathsf{good}] + \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} = i \mid \mathsf{bad}\right] \cdot \Pr[\mathsf{bad}] \\
&\leq \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} = i \mid \mathsf{good}\right] + \Pr[\mathsf{bad}] \leq \Pr\left[a_i = s \wedge b_i = s\right] + \mu(k) \\
&\leq \Pr[a_i = s] + \mu(k) \qquad (4)
\end{aligned}$$

Let $r(k) = E[R^{\Gamma}_{(\sigma_1,\sigma_2)}]$. Then, by Markov's inequality it holds that:

$$\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} \geq 2r(k)\right] \leq \frac{1}{2}$$

and thus:

$$\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} < 2r(k)\right] > \frac{1}{2}$$

We compute an upper bound for the event $R^{\Gamma}_{(\sigma_1,\sigma_2)} < 2r(k)$, using Eq. (4):

$$\begin{aligned}
\frac{1}{2} &< \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} < 2r(k)\right] = \Pr\left[\bigvee_{i=0}^{2r(k)-1} R^{\Gamma}_{(\sigma_1,\sigma_2)} = i\right] \leq \sum_{i=0}^{2r(k)-1} \Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} = i\right] \\
&\leq \sum_{i=0}^{2r(k)-1} \left(\Pr\left[a_i = s\right] + \mu(k)\right) \leq \sum_{i=0}^{2r(k)-1} \Pr\left[a_i = s\right] + 2r(k)\mu(k)
\end{aligned}$$

Therefore, we have:

$$\sum_{i=0}^{2r(k)-1} \Pr\left[a_i = s\right] > \frac{1}{2} - 2r(k)\mu(k)$$

On the other hand, using Claim 4.2:

$$\begin{aligned}
\sum_{i=0}^{2r(k)-1} \Pr\left[a_i = s\right] &\leq \sum_{i=0}^{2r(k)-1} \left(2i\beta + \mu(k)\right) = 2\beta \cdot \sum_{i=0}^{2r(k)-1} i + \sum_{i=0}^{2r(k)-1} \mu(k) \\
&\leq 8r^2(k)\beta + 2r(k)\mu(k)
\end{aligned}$$

Combining the above together we have:

$$\frac{1}{2} - 2r(k)\mu(k) < \sum_{i=0}^{2r(k)-1} \Pr[a_i = s] \leq 8r^2(k)\beta + 2r(k)\mu(k)$$

That is:

$$\frac{1}{2} < 8r^2(k)\beta + 4r(k)\mu(k)$$

Since β is a fraction of polynomials, and μ is a negligible function, for sufficiently large k's it holds that:

$$\beta(k) > \mu(k)$$

and therefore:

$$\frac{1}{2} < 8r^2(k)\beta + 4r(k)\mu(k) < 8r^2(k)\beta + 4r(k)\beta < 32r^2(k)\beta$$

We conclude that:

$$\begin{aligned} 32r^2(k)\beta &> \frac{1}{2} \\ r^2(k) &> \frac{1}{64\beta} \\ r(k) &> \frac{1}{8\sqrt{\beta}} \end{aligned}$$

This completes the proof. ■

Conclusion: Using Theorem 4.3 we conclude that there do not exist U^+-independent fair reconstruction mechanisms with an expected number of rounds that is polynomial, even in the simultaneous model. In order to see this, we show that for all fixed polynomials U_i, U_i^-, U_i^{--} and $r(k)$, there exists a polynomial U_i^+ such that $r(k) \leq \frac{1}{8\sqrt{\beta}}$. Specifically, take $U_i^+ \geq 64r^2(k) \cdot \left(U_i - U_i^{--}\right) + U_i^{--}$. This suffices because in such a case

$$\beta_i = \frac{U_i - U_i^{--}}{U_i^+ - U_i^{--}} \leq \frac{U_i - U_i^{--}}{64r^2(k) \cdot \left(U_i - U_i^{--}\right) + U_i^{--} - U_i^{--}} = \frac{U_i - U_i^{--}}{64r^2(k) \cdot \left(U_i - U_i^{--}\right)} = \frac{1}{64r^2(k)}$$

and thus $r(k) \leq \frac{1}{8\sqrt{\beta_i}}$ in contradiction. We have therefore proven the following:

Theorem 4.4 *There do not exist U^+-independent fair reconstruction mechanisms for two parties, even in the simultaneous channels model.*

4.2 Impossibility for U^f-Independence (Non-Simultaneous)

In Section 3 we showed that any mechanism that is U^f-independent achieves correctness. In the simultaneous channels model, U^f-independence – and correctness – has been achieved by previous protocols [6, 11]. However, as we have mentioned, the known protocols for the model with non-simultaneous channels do *not* guarantee correctness. In particular, if $U_i^f > U_i$ for some party P_i then the strategy profiles $\vec{\sigma}$ of [12, 8] are *not* computational Nash equilibriums. In this section we prove that this is inherent to the non-simultaneous model. That is, there does not exist a fair reconstruction mechanism that is U^f-independent in the non-simultaneous model.

The Kol-Naor mechanism [12] and correctness. Before proceeding with our proof, we describe the mechanism of Kol and Naor for non-simultaneous channels and show why it does not achieve correctness. This example illustrates the problem of achieving U^f-independence and is thus very instructive. The Kol-Naor mechanism assumes that the utility functions U^+, U and U^- fulfill the assumptions in Definition 2.5. Furthermore, the mechanism itself is constructed given the actual values of the utility functions (i.e., it is utility dependent). The general idea of their protocol is that the shares assigned to the party are actually lists of possible secrets. One party receives a list of size ℓ (this party is called "the short party"), and the other party receives a list of size $\ell + d$ (this party is called "the long party"). The short list is a strict prefix of the other. The lengths ℓ and d are chosen according to a geometric distribution with parameter β, where β depends on the utility functions of the parties. The real secret is located at position $\ell + 1$ in the long list, while all the other elements in the lists are fake; the $(\ell + 1)$th round is called the definitive round because in this round the secret is learned. In addition to the lists described above, the dealer selects an independent random permutation for every round; this permutation determines the order in which the parties send their list elements in the round. The party that sends its message first in the definitive round is given the long list, and the other party is given the short list. In addition, the parties receive the permutations for the rounds appearing in their respective lists (i.e., the short party receives the permutation only for the first ℓ rounds). We stress that neither party knows if it the short or long party. In any given round, we call the party who sends its element first the "first party" and we call the other the "second party".

In order to reconstruct the secret, the parties proceed round by round; in the ith round each party sends its ith list element in the order determined by the permutation. At iteration $\ell + 1$ (the "definitive iteration"), the long party is the first to broadcast its share (that is, it is the "first party"). However, the short party's list is finished and thus it has no element to send. It therefore remains silent in this round. The first round in which only one party sends a list element is the definitive round, and so the secret sent in this round is taken to be the real secret. Intuitively, fairness is achieved because the owner of the long list does not know the length of the short list, and in particular does not know which round is the definitive round. It therefore does not know which of the elements in its list is the real secret and so has to send its share every round. See [12] for details.

As pointed out in [12, Note 6.2], if one of the parties aborts prematurely (i.e., remains silent in round i for some $i < \ell$) then the other party will output an incorrect value (with high probability the element s_i of the ith round will not equal the secret). It is important to note that the aborting party knows that s_i is not the real secret because its list is not yet finished. Furthermore, it can even have some influence over the incorrect value output by the first party (this is because it can choose at which point to stop and thus it can choose which of the values in the prefix of the list is output by the first party). The protocol is therefore clearly not correct. We remark that the same problem also exists for the protocol of [8]. As we have mentioned, [12] assume that rational parties will not behave in this way because they always prefer to learn the secret than to not learn it (observe that if a party aborts prematurely then it will not learn the real secret). That is, they assume that $U_i^f < U_i$. We show that this assumption is essential as long as U^f-independence is desired.

The impossibility result. Our proof of impossibility assumes that for all i, U_i^+ is strictly greater than U_i by a non-negligible amount. This is called strict competitiveness [12] and is defined as follows:

Definition 4.5 (strict competitiveness): *Let* $\mathcal{U} = \{(U_i^+, U_i, U_i^-, U_i^{--}, U^f)_{i\in\{1,2\}}\}$ *be a set of natural utility functions. We say that the set is* strictly competitive *if for every* $i \in \{1,2\}$*, there exists a polynomial* $p(\cdot)$*, such that for infinitely many* k*'s it holds that:*

$$U_i^+(k) \geq U_i(k) + \frac{1}{p(k)}$$

We are now ready to formally state the theorem.

Theorem 4.6 *For any set of strictly competitive utility functions* $\mathcal{U}$*, there does not exist a fair reconstruction mechanism that is* U^f*-independent in the non-simultaneous model.*

By Proposition 3.4, U^f-independence implies correctness. We therefore prove that in the non-simultaneous model there does not exist a fair reconstruction mechanism that is *correct*, as defined in Definition 3.2.

Intuition: We begin by describing 2 strategies σ_1^{stop} and σ_2^{stop}. The strategy σ_1^{stop} for party P_1 is the strategy that follows the prescribed strategy $\vec{\sigma}$ in all the rounds with the following difference. In every round, P_1 checks what its output would be if P_2 quits at that round. In the first round for which the output is *not* $\perp$, the strategy σ_1^{stop} instructs P_1 to quit at that round. σ_2^{stop} is defined analogously. Since we assume correctness, the probability that one of the parties will output a value which is not s or $\perp$ when the other prematurely aborts is negligible. Thus, when playing $\vec{\sigma}^{stop}$ both of the parties will output the correct s in the round that they quit. Next, we prove that when both parties follow $\vec{\sigma}^{stop}$, with high probability one of them learns the secret while the other does not. We conclude by showing that the prescribed strategy $\vec{\sigma}$ is not a computational Nash equilibrium by showing that one of the $\vec{\sigma}^{stop}$ strategies has a better expected utility than $\vec{\sigma}$. That is, we show that either $u_2(\sigma_1, \sigma_2^{stop}) > u_2(\sigma_1, \sigma_2) + \epsilon'$ or $u_1(\sigma_1^{stop}, \sigma_2) > u_1(\sigma_1, \sigma_2) + \epsilon'$, for some non-negligible function ϵ'.

Before proceeding to the formal proof, we introduce the necessary notation.

Notations and conventions: Assume that P_1 sends the first message, and P_2 sends the last message. A *round* of $(\Gamma, \vec{\sigma})$ consists of a message from P_1 followed by a message from P_2. If P_1 aborts before sending its ith-round message, then we will say that P_2 outputs b_{i-1} (thus, if P_1 does not send any messages then P_2 outputs b_0). Furthermore, if P_2 aborts before sending its ith-round message (which is after it has received the ith message from P_1), then we say that P_1 outputs a_i. Thus, if P_2 sends no messages, P_1 outputs a_1. For simplicity, if party P_1 halts at round i (and outputs a_i), we define $a_j = a_i$ for every $j > i$. We define $b_j = b_i$ after P_2 halts in the same way. We call these values "the default output". See Figure 2.

Random coins. Denote by $\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma,\rho_1),P_2(\sigma,\rho_2)}\left(\text{SHARE}\left(\mathcal{S},\rho_s\right)\right)\right)$ the exact number of rounds of the execution of the game Γ with parties P_1 and P_2 using random coins ρ_1 and ρ_2, respectively, where both parties follow the prescribed strategy $\vec{\sigma}$ and use the shares generated by the algorithm SHARE with coins ρ_s (recall that ρ_s also determines the secret s chosen according to $\mathcal{S}$). Furthermore, denote by $\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma),P_2(\sigma)}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right)$ the random variable on the number of rounds when ρ_1, ρ_2, ρ_s are uniformly distributed. We denote the expected number of rounds by $r(k)$; that is $r(k) = E\left[\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma),P_2(\sigma)}\left(\text{SHARE}\left(\mathcal{S}\right)\right)\right)\right]$. For the sake of clarity

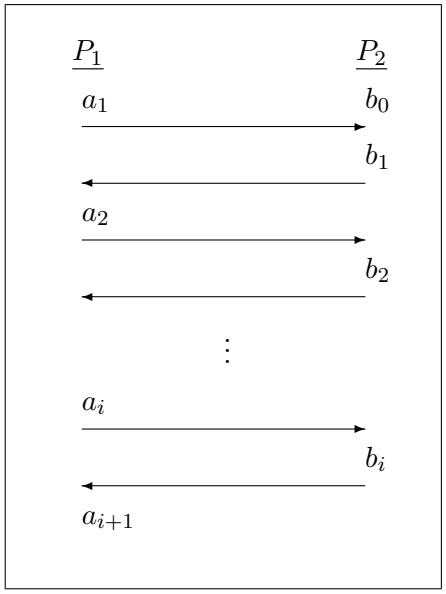


Figure 2: Default output notations (non-simultaneous)

(and to reduce the amount of notation), we include the sampling of s according to $\mathcal{S}$ in the random coins of the algorithm SHARE used to generate the shares.

Proof of Theorem 4.6: Let $\mathcal{U}$ be a set of strictly competitive utility functions and let $\mathcal{S}$ be the uniform distribution over $\{0,1\}^n$. We prove that for every mechanism $(\Gamma, \vec{\sigma})$ one of the following does not hold:

1. $(\Gamma, \vec{\sigma})$ is a fair reconstruction mechanism for the set of utility functions $\mathcal{U}$.
2. $(\Gamma, \vec{\sigma})$ provides correctness. That is, for *every arbitrary strategy* σ_i' followed by party P_i ($i \in \{1,2\}$) there exists a negligible function μ such that:

$$\Pr[\text{OUTPUT}_{P_{-i}}(\text{REAL}_{\Gamma,P_i(\sigma_i'),P_{-i}(\sigma_{-i})}(\text{SHARE}(\mathcal{S}))) \notin \{\mathcal{S}, \perp\}] \leq \mu(k).$$

It suffices to show that every protocol that achieves correctness (item 2) is not a fair reconstruction mechanism (item 1). We stress that we ignore the U^f utilities in this proof because we assume U^f-independence.

In the proof below, we will define a set Ω of "good" coins for (SHARE, P_1, P_2). The set contains all of the coins with the property that an execution of the share (SHARE) and reconstruction protocol ($(\Gamma, \vec{\sigma})$) with these coins fulfill a number of requirements. We will then show that the size of the set Ω is "large". Furthermore, we show that when both parties use these coins and follow $\vec{\sigma}^{stop}$, then the outcome must be o_1^+ or o_2^+ (i.e., only one of the parties learns the output). Finally, based on this, we show that (σ_1, σ_2) is not a computational Nash Equilibrium, since for at least one of the parties P_i, its utility is larger by a non-negligible amount when following σ_i^{stop} than when following σ_i.

Definition 4.7 (the set of coins $\Omega_{p(k)}$) *Let $p(\cdot)$ be a polynomial. Define the set $\Omega_{p(k)} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ to be the set of coins (ρ_s, ρ_1, ρ_2) for (SHARE, P_1, P_2) that fulfill the following requirements for parties running the prescribed strategy $\vec{\sigma}$:*

- req_1*: The number of rounds in the execution of the reconstruction protocol is less than* $p(k)$*; that is* $\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma,\rho_1),P_2(\sigma,\rho_2)}\left(\text{SHARE}\left(\mathcal{S},\rho_s\right)\right)\right) < p(k)$
- req_2*: In every round, the default outputs of both parties are in* $\{s,\perp\}$*. That is:*
  - *For every* i $(1 \leq i \leq p(k))$*, it holds that* $a_i \in \{s,\perp\}$*.*
  - *For every* j $(0 \leq j \leq p(k)-1)$*, it holds that* $b_j \in \{s,\perp\}$*.*
- req_3*: Neither party outputs the correct value* s *without participating in the protocol. That is* $a_1 \neq s$ *and* $b_0 \neq s$*.*
- req_4*: Both parties output* s *at the end of the protocol. That is,* $a_{p(k)-1} = s$ *and* $b_{p(k)} = s$*.*

We stress that the set $\Omega_{p(k)}$ is well defined because the parties' strategies are specified and thus the random coins fully determine the flow of execution and outputs. Recall that $r(k)$ is the expected number of rounds in the reconstruction protocol. The following claim states that "most" of the random coins are in the set $\Omega_{p(k)}$:

Claim 4.8 *For every polynomial* $q(\cdot)$*, every polynomial* $p(k) \geq 2 \cdot r(k) \cdot q(k)$*, and all sufficiently large* k*'s, it holds that*

$$\Pr\left[\rho \in \Omega_{p(k)}\right] \geq 1 - \frac{1}{q(k)}$$

where $\rho \in_R \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ *is uniformly distributed.*

Proof: Let $q(\cdot)$ be a polynomial, and define $p(k) \geq 2 \cdot r(k) \cdot q(k)$. Now, let $\rho \in_R \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be uniformly distributed. We calculate the probability that $\rho \in \Omega_{p(k)}$, which occurs when all the requirement of Definition 4.7 are met by ρ. That is:

$$\begin{aligned}\Pr\left[\rho \in \Omega_{p(k)}\right] &= \Pr\left[\mathsf{req}_1 \wedge \mathsf{req}_2 \wedge \mathsf{req}_3 \wedge \mathsf{req}_4\right] \\ &= \Pr\left[\mathsf{req}_1\right] \cdot \Pr\left[\mathsf{req}_2 \mid \mathsf{req}_1\right] \cdot \Pr\left[\mathsf{req}_3 \mid \mathsf{req}_1 \wedge \mathsf{req}_2\right] \cdot \Pr\left[\mathsf{req}_4 \mid \mathsf{req}_1 \wedge \mathsf{req}_2 \wedge \mathsf{req}_3\right]\end{aligned}$$

Computing the first term $\mathbf{Pr[req_1]}$: The first requirement states that the number of rounds should be no more than $p(k)$. Since the expected number of rounds is $r(k)$, by Markov's inequality we have that

$$\Pr\left[\neg\mathsf{req}_1\right] = \Pr\left[\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma,\rho_1),P_2(\sigma,\rho_2)}\left(\text{SHARE}\left(\mathcal{S},\rho_s\right)\right)\right) \geq p(k)\right] \leq \frac{r(k)}{p(k)}$$

and thus:

$$\Pr\left[\mathsf{req}_1\right] = \Pr\left[\text{ROUNDS}\left(\text{REAL}_{\Gamma,P_1(\sigma,\rho_1),P_2(\sigma,\rho_2)}\left(\text{SHARE}\left(\mathcal{S},\rho_s\right)\right)\right) < p(k)\right] \geq 1 - \frac{r(k)}{p(k)}$$

Computing the second term $\mathbf{Pr[req_2 \mid req_1]}$: Given that req_1 holds, we know that the protocol terminates after no more than $p(k)$ rounds. That is, it suffices to show that for every $i \in \{1,...,p(k)\}$, $a_i \in \{s,\perp\}$, and for every $j \in \{0,...,p(k)-1\}$, $b_j \in \{s,\perp\}$. Thus,

$$\begin{aligned}\Pr\left[\mathsf{req}_2 \mid \mathsf{req}_1\right] &= \Pr\left[(\forall i \in \{1,...,p(k)\}, a_i \in \{s,\perp\}) \wedge (\forall j \in \{0,...,p(k)-1\}, b_j \in \{s,\perp\})\right] \\ &= 1 - \Pr\left[(\exists i \in \{1,...,p(k)\}, a_i \notin \{s,\perp\}) \vee (\exists j \in \{0,...,p(k)-1\}, b_j \notin \{s,\perp\})\right]\end{aligned}$$

Using the union bound, we get:

$$\begin{aligned}
&\Pr\left[(\exists i \in \{1,...,p(k)\}, a_i \notin \{s,\perp\}) \vee (\exists j \in \{0,...,p(k)-1\}, b_j \notin \{s,\perp\})\right] \\
&\quad \leq \sum_{i=1}^{p(k)} \Pr\left[a_i \notin \{s,\perp\}\right] + \sum_{j=0}^{p(k)-1} \Pr\left[b_j \notin \{s,\perp\}\right] \\
&\quad \leq 2p(k) \cdot \mu(k)
\end{aligned}$$

for some negligible function μ. This last inequality follows from the assumption that the reconstruction protocol achieves *correctness* and thus a party outputs an incorrect value with only negligible probability (even if the other party terminates early). By the fact that $p(\cdot)$ is a polynomial, it follows that there exists a negligible function μ_1 such that

$$\Pr\left[\mathsf{req}_2 \mid \mathsf{req}_1\right] \geq 1 - 2p(k) \cdot \mu(k) \geq 1 - \mu_1(k)$$

Computing the third term $\mathbf{Pr[req_3 \mid req1 \wedge req_2]}$: The third requirement states that neither party outputs s without participating in the protocol. Using req_2, we know that $a_1 \in \{s,\perp\}$, $b_0 \in \{s,\perp\}$ and thus:

$$\Pr\left[\mathsf{req}_3 \mid \mathsf{req}_1 \wedge \mathsf{req}_2\right] = \Pr\left[a_1 = \perp \wedge b_0 = \perp \mid a_1, b_0 \in \{s,\perp\}\right]$$

We calculate:

$$\begin{aligned}
\Pr\left[\mathsf{req}_3 \mid \mathsf{req}_1 \wedge \mathsf{req}_2\right] &= 1 - \Pr\left[a_1 = s \vee b_0 = s \mid a_1, b_0 \in \{s,\perp\}\right] \\
&\geq 1 - \Pr\left[a_1 = s \mid a_1, b_0 \in \{s,\perp\}\right] - \Pr\left[b_0 = s \mid a_1, b_0 \in \{s,\perp\}\right]
\end{aligned}$$

We show that $\Pr\left[a_1 = s \mid a_1, b_0 \in \{s,\perp\}\right]$ is negligible (the calculation for b_0 is equivalent). Since $\mathcal{S}$ is the uniform distribution over $\{0,1\}^k$, it holds that:

$$\begin{aligned}
\Pr\left[a_1 = s \mid a_1, b_0 \in \{s,\perp\}\right] &\leq \frac{\Pr\left[a_1 = s\right]}{\Pr\left[a_1, b_0 \in \{s,\perp\}\right]} \\
&= \frac{2^{-k}}{\Pr\left[a_1, b_0 \in \{s,\perp\}\right]} \\
&\leq \frac{2^{-k}}{1-\mu(k)} \leq \mu_2(k)
\end{aligned}$$

for some negligible function $\mu_2(k)$. (The equality follows from the fact that $\mathcal{S}$ is the uniform distribution over $\{0,1\}^k$ and the last inequality follows from the assumption of correctness on the protocol.) Thus,

$$\Pr\left[\mathsf{req}_3 \mid \mathsf{req}_1 \wedge \mathsf{req}_2\right] \geq 1 - 2\mu_2(k)$$

Computing the fourth term $\mathbf{Pr[req_4 \mid req1 \wedge req_2 \wedge req_3]}$: The fourth requirement says that at the end of the execution of the protocol, both parties output the secret s. By the *completeness* of the protocol when both parties follow the prescribed strategy, we know that whenever the strategy instructs the parties to halt, they must output s, except with negligible probability. Since we

are conditioning on req_1, the prescribed strategy instructs the parties to halt before round $p(k)$. Therefore, there exists a negligible function μ_3 such that

$$\Pr\left[\mathsf{req}_4 \mid \mathsf{req}_1 \wedge \mathsf{req}_2 \wedge \mathsf{req}_3\right] \geq 1 - \mu_3(k)$$

Completing the proof: Combining all of the above we have that for all sufficiently large k's:

$$\begin{aligned}
\Pr\left[\rho \in \Omega_{p(k)}\right] &= \Pr\left[\mathsf{req}_1\right] \cdot \Pr\left[\mathsf{req}_2 \mid \mathsf{req}_1\right] \cdot \Pr\left[\mathsf{req}_3 \mid \mathsf{req}_1 \wedge \mathsf{req}_2\right] \cdot \Pr\left[\mathsf{req}_4 \mid \mathsf{req}_1 \wedge \mathsf{req}_2 \wedge \mathsf{req}_3\right] \\
&\geq \left(1 - \frac{r(k)}{p(k)}\right) \cdot (1 - \mu_1(k)) \cdot (1 - 2\mu_2(k)) \cdot (1 - \mu_3(k)) \\
&\geq 1 - \frac{r(k)}{p(k)} - \frac{r(k)}{p(k)}
\end{aligned}$$

where the last inequality is due to the fact that $\frac{r(k)}{p(k)}$ is larger than any negligible function (for all sufficiently large k's). Since $p(k) \geq 2 \cdot r(k) \cdot q(k)$:

$$\Pr\left[\rho \in \Omega_{p(k)}\right] \geq 1 - \frac{2 \cdot r(k)}{p(k)} \geq 1 - \frac{2 \cdot r(k)}{2 \cdot r(k) \cdot q(k)} = 1 - \frac{1}{q(k)}$$

This completes the proof of Claim 4.8. ■

We now formally define the strategy $\vec{\sigma}^{stop}$ for both parties:

Formal description of $\boldsymbol{\sigma_1^{stop}}$:

- $i \leftarrow 1$.
- calculate a_1.
- While $a_i = \perp$
  - Run P_1 for round i (send a message using the strategy σ_1 and then receive a message).
  - Increment i.
  - Calculate a_i.
- On the first element such that $a_i \neq \perp$, quit, and output a_i.

Formal description of $\boldsymbol{\sigma_2^{stop}}$:

- $j \leftarrow 0$.
- calculate b_0.
- While $b_j = \perp$
  - Increment j.
  - Receive the round j message from P_1.
  - Calculate b_j.
  - If $b_j = \perp$, run P_2 for round j (send a message using the strategy σ_2).
- On the first element such that $b_j \neq \perp$, quit, and output b_j.

We remark that for the analysis to go through, it is crucial that the strategy $\vec{\sigma}^{stop}$ uses the exact same random coins as $\vec{\sigma}$. Thus, if $\vec{\sigma}$ does not calculate a_i or b_j in every step (since such a value is only needed if indeed the other party aborts early), and if these calculations are probabilistic and require random coins, then $\vec{\sigma}$ and $\vec{\sigma}^{stop}$ need a different number of random coins. In order to overcome this technical detail, we define the profile strategy $\vec{\sigma}^{equal}$ which is equivalent to $\vec{\sigma}$ except that it internally calculates a_i b_j in every step (and does nothing with this additional information). It is easy to see that $\vec{\sigma}$ and $\vec{\sigma}^{equal}$ are equivalent. That is, if $\vec{\sigma}$ is a computational Nash Equilibrium, then so is $\vec{\sigma}^{equal}$. Moreover, $\vec{\sigma}^{equal}$ and $\vec{\sigma}^{stop}$ use the exact same number of coins. For the sake of

clarity, we ignore this from now on and just assume that $\vec{\sigma}$ itself uses the same number of coins as $\vec{\sigma}^{stop}$

We have shown that for every polynomial, there is a large set of random coins $\Omega_{p(k)}$ that result in a "good" outcome in an execution of the strategy $\vec{\sigma}$. We now show that for every set of random tapes ρ in $\Omega_{p(k)}$, if both parties change their strategy to $\vec{\sigma}^{stop}$ and use such a ρ, then with probability 1 *exactly one of the parties* will learn the secret. We stress that this is guaranteed only for the "good" coins in $\Omega_{p(k)}$ and thus the probability overall that this occurs when both parties follow $\vec{\sigma}^{stop}$ equals the probability that a set of random coins are in $\Omega_{p(k)}$ (which is close to 1). Before stating the claim, we introduce the following shorthand. We write

$$\left[o \mid (\sigma_1', \sigma_2'', \rho \in \Omega_{p(k)})\right]$$

to denote the event that the outcome is o after an execution where party 1 runs strategy σ_1', party 2 runs strategy σ_2'' and the coins used, ρ, are in the set $\Omega_{p(k)}$ as defined in Definition 4.7. Recall also that o_i^+ is the outcome that P_i learns the secret while P_{-i} does not. We have:

Claim 4.9 *For every polynomial $p(\cdot)$ it holds that:*

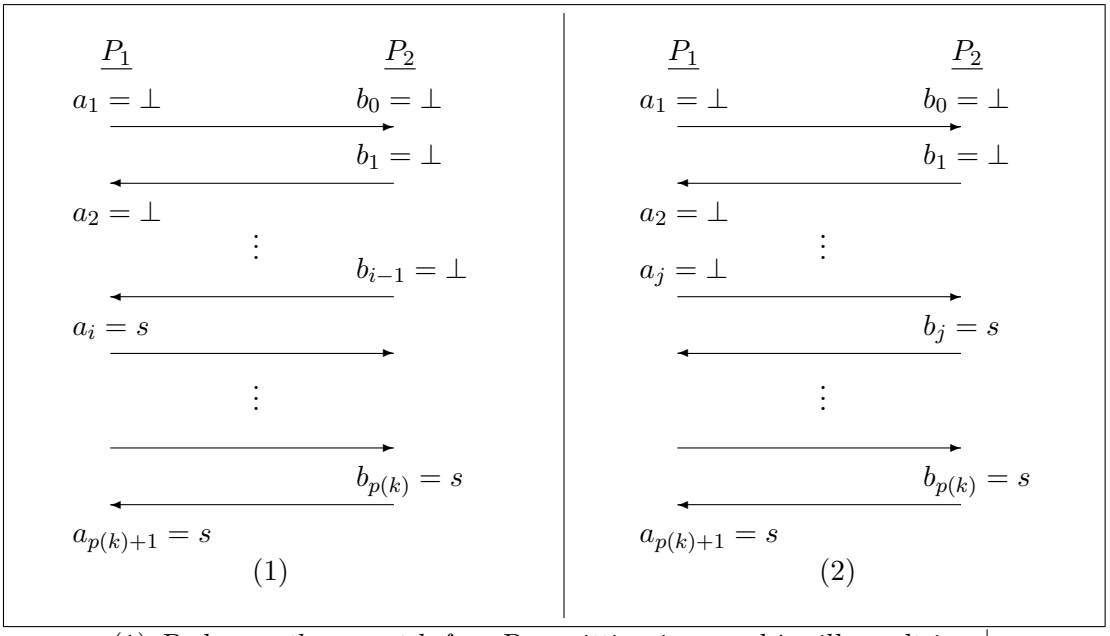
$$\Pr\left[o_1^+ \vee o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}, \rho \in \Omega_{p(k)})\right] = 1$$

Proof: Let $p(\cdot)$ be a polynomial. By the definition of $\Omega_{p(k)}$ we have that for every $\rho \in \Omega_{p(k)}$, the following holds:

- The number of rounds is less than $p(k)$.
- At the end of the protocol, both parties learn the secret (as long as they follow the prescribed strategy).
- $a_1 = \perp$, $b_0 = \perp$.
- For every $i \in \{1, ..., p(k)\}$, $j \in \{0, ..., p(k)-1\}$, $a_i \in \{s, \perp\}$, $b_j \in \{s, \perp\}$.

It follows that when both of the parties follow $\vec{\sigma}$, there exists a round i such that $a_i \neq \perp$, and a round j such that $b_j \neq \perp$ (this must occur since $a_1 = b_0 = \perp$ and $a_{p(k)} = b_{p(k)-1} = s$). We will consider the first i such that $a_i \neq \perp$ and the first j such that $b_j \neq \perp$. Since all a_i, b_j values are in $\{s, \perp\}$ it follows that $a_i = s$ and $b_j = s$. Now, if both parties will play according to $\vec{\sigma}^{stop}$, then when one of them receives a value which is not $\perp$ it will immediately stop. Since we are working in the non-simultaneous model, we can show that:

- If $i \leq j$, then by the definition of σ_1^{stop} party P_1 will not send its round i message and party P_2 will output $\perp$. (When $i < j$ this is clear. Furthermore, when $i = j$, note that P_1 defines a_i before sending its message in round i and P_2 defines b_j only after receive this message from P_1. Thus, when $i = j$, P_2 will output $b_{j-1} = \perp$.) In this case, P_1 outputs s and P_2 outputs $\perp$; thus the outcome is o_1^+. See Figure 3, case (1).
- If $j < i$, then by the definition of σ_2^{stop} party P_2 will stop before sending its message in round j. Thus, $a_j = \perp$. This implies that P_2 alone learns the secret (since $b_j = s$) and P_1 outputs $\perp$. The outcome in this case is o_2^+. See Figure 3, case (2).



(1) *P_1 learns the secret before P_2; quitting in round i will result in o_1^+*
(2) *P_2 learns the secret before P_1; quitting in round j will result in o_2^+*

Figure 3: The possible cases of a run with $\rho \in \Omega_{p(k)}$

We conclude that when $\rho \in \Omega_{p(k)}$ and both parties follow $\vec{\sigma}^{stop}$, the probability that exactly one of the parties learns the secret is 1, as required. ■

In the previous claim we showed that when both parties play according to $\vec{\sigma}^{stop}$, and when $\rho \in \Omega_{p(k)}$, there will be a party that learns the secret alone.

Our aim overall is to show that the strategy $\vec{\sigma}$ does not achieve computational Nash equilibrium. In order to do this, we show that for a particular party, there exists a better strategy; in particular, we will show that σ^{stop} is better for one of the parties. A key step in doing this is the next technical claim which considers what happens when one party P_i uses σ_i^{stop} while the other uses σ_{-i}. We use the previous claim to show that:

Corollary 4.10 *For every polynomial $p(\cdot)$ it holds that:*

$$\Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] + \Pr\left[o_2^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] = 1$$

Proof: In Claim 4.9 we proved that:

$$\Pr\left[o_1^+ \vee o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] = 1$$

We observe the following three facts:

1. o_1^+ and o_2^+ are disjoint events (i.e., they cannot both occur in a single execution). Hence:

$$\begin{aligned}&\Pr\left[o_1^+ \vee o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \\ &\quad = \Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] + \Pr\left[o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right]\end{aligned}$$

2. We claim that:

$$\Pr[o_1^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}] \leq \Pr[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}].$$

   This is true since in every run with the parties following $\left(\sigma_1^{stop}, \sigma_2^{stop}\right)$ in which P_1 learns the secret before P_2, it follows that P_1 learns the secret before P_2 even if P_2 plays according to σ_2. (Until the point that P_1 learns the secret, σ_2 and σ_2^{stop} are identical.)

3. $\Pr[o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}] \leq \Pr[o_2^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}]$. This follows as above.

Combining the above, we conclude that:

$$\begin{aligned}&\Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] + \Pr\left[o_2^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \\ &\quad \geq \Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] + \Pr\left[o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \\ &\quad = \Pr\left[o_1^+ \vee o_2^+ \mid (\sigma_1^{stop}, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] = 1\end{aligned}$$

completing the proof of the claim. ■

We are now ready to prove our main claim:

Claim 4.11 *The prescribed strategy $\vec{\sigma} = (\sigma_1, \sigma_2)$ is not a computational Nash equilibrium.*

Proof: We prove this by showing that there exists a *non-negligible* function $\epsilon' = \epsilon'(k)$ for which at least one of the following holds:

$$u_1(\sigma_1^{stop}, \sigma_2) > u_1(\sigma_1, \sigma_2) + \epsilon'$$

$$u_2(\sigma_1, \sigma_2^{stop}) > u_2(\sigma_1, \sigma_2) + \epsilon'$$

Let $q(\cdot)$ be a polynomial such that for all sufficiently large k's it holds that:

$$q(k) \geq \max\left\{\frac{2U_1^+ + 2U_1}{U_1^+ - U_1}, \frac{2U_2^+ + 2U_2}{U_2^+ - U_2}\right\} \tag{5}$$

Such a polynomial $q(\cdot)$ exists because all of the utility functions are polynomial, and by strict competitiveness there is a non-negligible difference between U_i^+ and U_i. Now, let $p(k) \geq 2 \cdot r(k) \cdot q(k)$ (where $r(k)$ equals the expected number of rounds and $q(k)$ is as above), and define $\Omega_{p(k)}$ as in Definition 4.7, using this $p(k)$.

Notation:

- Denote by α the probability that party P_1 learns the secret and P_2 does not, when P_1 follows σ_1^{stop}, P_2 follows σ_2, and $\rho \in \Omega_{p(k)}$. That is:

$$\alpha = \Pr[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}]$$

- Denote by β the probability that $\rho \in \Omega_{p(k)}$. Recall that by Claim 4.8:

$$\beta = \Pr[\rho \in \Omega_{p(k)}] \geq \left(1 - \frac{1}{q(k)}\right)$$

The expected utility of P_1: The expected utility of party P_1 when it plays according to σ_1^{stop}, P_2 plays according to σ_2, and when $\rho \in \Omega_{p(k)}$ is:

$$\begin{aligned} u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \in \Omega_{p(k)}\right) &= \Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^+ + \Pr\left[o^{\text{both}} \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1 \\ &\quad + \Pr\left[o_1^- \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^- + \Pr\left[o_2^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^{--} \end{aligned}$$

We now calculate each of the above probabilities:

- By our above notation, $\Pr\left[o_1^+ \mid \left(\sigma_1^{stop}, \sigma_2\right), \rho \in \Omega_{p(k)}\right] = \alpha$.
- $\Pr\left[o_1^- \mid \left(\sigma_1^{stop}, \sigma_2\right), \rho \in \Omega_{p(k)}\right] = 0$: This holds because when P_2 plays according to σ_2 we know that P_1 will always learn the secret. Recall that $\rho \in \Omega_{p(k)}$ and so when neither party stops early, both output s with probability 1. Since P_2 does not stop early, P_1 will only stop when it learns s (note that P_1 only stops when $a_i \neq \perp$ and by req_2 we are given that $a_i = s$).
- $\Pr\left[o_2^+ \mid \left(\sigma_1^{stop}, \sigma_2\right), \rho \in \Omega_{p(k)}\right] = 0$: This follows from the exact same argument as above.
- $\Pr\left[o^{\text{both}} \mid \left(\sigma_1^{stop}, \sigma_2\right), \rho \in \Omega_{p(k)}\right] = 1 - \alpha$: This holds because the sum of all four probabilities in the expected utility of P_1 equals 1 (they cover all possibilities).

We therefore have

$$u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \in \Omega_{p(k)}\right) = \alpha \cdot U_1^+ + (1 - \alpha) \cdot U_1$$

We have calculated the expected utility of P_1 for $\rho \in \Omega_{p(k)}$. We conclude by calculating the expected utility for P_1 for a *randomly chosen* ρ:

$$\begin{aligned} u_1\left(\sigma_1^{stop}, \sigma_2\right) &= u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \in \Omega_{p(k)}\right) \cdot \Pr\left[\rho \in \Omega_{p(k)}\right] + u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \notin \Omega_{p(k)}\right) \cdot \Pr\left[\rho \notin \Omega_{p(k)}\right] \\ &\geq u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \in \Omega_{p(k)}\right) \cdot \Pr\left[\rho \in \Omega_{p(k)}\right] \\ &= \left(\alpha \cdot U_1^+ + (1 - \alpha) \cdot U_1\right) \cdot \beta \end{aligned} \tag{6}$$

Below, we will show that when $\alpha \geq 1/2$ (and for an appropriate β), this expected utility is greater than $U_1 + \epsilon'$, where ϵ' is non-negligible. Before doing this, we carry out an analogous calculation for the expected utility of P_2.

The expected utility of P_2: The expected utility of party P_2 when it follows σ_2^{stop}, P_1 follows σ_1, and $\rho \in \Omega_{p(k)}$ is:

$$\begin{aligned} u_2\left(\sigma_1, \sigma_2^{stop} \mid \rho \in \Omega_{p(k)}\right) &= \Pr\left[o_2^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \cdot U_2^+ + \Pr\left[o^{\text{both}} \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \cdot U_2 \\ &\quad + \Pr\left[o_2^- \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \cdot U_2^- + \Pr\left[o_1^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] \cdot U_2^{--} \end{aligned}$$

We calculate each of the above probabilities:

- $\Pr[o_2^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}] = 1 - \alpha$. This follows from Claim 4.10 and the fact that $\Pr\left[o_1^+ \mid \left(\sigma_1^{stop}, \sigma_2\right), \rho \in \Omega_{p(k)}\right] = \alpha$.
- $\Pr[o_2^- \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}] = 0$. This is exactly the same as in the calculation of the utility of P_1.
- $\Pr\left[o_1^+ \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] = 0$. Again, as above.
- $\Pr[o^{\text{both}} \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}] = \alpha$. Once again, this follows because all four probabilities sum to 1.

We therefore write:

$$u_2\left((\sigma_1, \sigma_2^{stop}) \mid \rho \in \Omega_{p(k)}\right) = (1-\alpha) \cdot U_2^+ + \alpha \cdot U_2$$

We have calculated the expected utility for P_2 when $\rho \in \Omega_{p(k)}$. The expected utility for P_2 for a randomly chosen ρ is:

$$\begin{aligned} u_2\left(\sigma_1, \sigma_2^{stop}\right) &= u_2\left(\sigma_1, \sigma_2^{stop} \mid \rho \in \Omega_{p(k)}\right) \cdot \Pr\left[\rho \in \Omega_{p(k)}\right] + u_2\left(\sigma_1, \sigma_2^{stop} \mid \rho \notin \Omega_{p(k)}\right) \cdot \Pr\left[\rho \notin \Omega_{p(k)}\right] \\ &\geq u_2\left(\sigma_1, \sigma_2^{stop} \mid \rho \in \Omega_{p(k)}\right) \cdot \Pr\left[\rho \in \Omega_{p(k)}\right] \\ &= \left((1-\alpha) \cdot U_2^+ + \alpha \cdot U_2\right) \cdot \beta \end{aligned} \tag{7}$$

Below, we will show that when $\alpha \leq 1/2$ (and for an appropriate β), this expected utility is greater than $U_1 + \epsilon'$, where ϵ' is non-negligible.

Computing β: In order to complete our analysis, we need to compute the value of β as a function of the utilities. Recall that we set

$$\beta = 1 - \frac{1}{q(k)}$$

and by Eq. (5):

$$q(k) \geq \max\left\{\frac{2U_1^+ + 2U_1}{U_1^+ - U_1}, \frac{2U_2^+ + 2U_2}{U_2^+ - U_2}\right\}$$

Let $i \in \{1,2\}$ be the index for which $\frac{2U_i^+ + 2U_i}{U_i^+ - U_i}$ is the maximum. We have:

$$\begin{aligned} \beta &= 1 - \frac{1}{q(k)} \geq 1 - \frac{1}{\frac{2U_i^+ + 2U_i}{U_i^+ - U_i}} = 1 - \frac{U_i^+ - U_i}{2U_i^+ + 2U_i} = \frac{2U_i^+ + 2U_i - U_i^+ + U_i}{2U_i^+ + 2U_i} \\ &= \frac{U_i^+ + 3U_i}{2U_i^+ + 2U_i} = \frac{4U_i + (U_i^+ - U_i)}{2U_i^+ + 2U_i} = \frac{U_i + \frac{1}{4}(U_i^+ - U_i)}{\frac{1}{2}U_i^+ + \frac{1}{2}U_i} \end{aligned}$$

Therefore, it holds that:

$$\beta \geq \frac{U_1 + \frac{1}{4}(U_1^+ - U_1)}{\frac{1}{2}U_1^+ + \frac{1}{2}U_1} \quad \text{and} \quad \beta \geq \frac{U_2 + \frac{1}{4}(U_2^+ - U_2)}{\frac{1}{2}U_2^+ + \frac{1}{2}U_2} \tag{8}$$

Concluding the proof: In order to show that $\vec{\sigma}$ is not a computational-Nash Equilibrium, we show that at least one of the parties P_i will prefer to change its strategy to $\vec{\sigma}_i^{stop}$. The specific party that will prefer to change its strategy depends on the value α. There are two cases:

- *Case 1 – $\frac{1}{2} \leq \alpha \leq 1$:* In this case, we claim that P_1 prefers to change its strategy to σ_1^{stop}. Recall that by Eq. (6):

$$u_1\left(\sigma_1^{stop}, \sigma_2\right) = \left(\alpha \cdot U_1^+ + (1-\alpha) \cdot U_1\right) \cdot \beta$$

Let $\alpha = \frac{1}{2} + \delta$, where $\delta \geq 0$. We write:

$$\begin{aligned} u_1\left(\sigma_1^{stop}, \sigma_2\right) &= \left(\left(\frac{1}{2}+\delta\right) \cdot U_1^+ + (1-\frac{1}{2}-\delta) \cdot U_1\right) \cdot \beta \\ &= \left(\left(\frac{1}{2}+\delta\right) \cdot U_1^+ + \left(\frac{1}{2}-\delta\right) \cdot U_1\right) \cdot \beta \\ &= \left(\delta \cdot \left(U_1^+ - U_1\right) + \frac{1}{2} \cdot U_1^+ + \frac{1}{2} \cdot U_1\right) \cdot \beta \\ &\geq \left(\frac{1}{2} \cdot U_1^+ + \frac{1}{2} \cdot U_1\right) \cdot \beta \end{aligned}$$

Using Eq. (8) we know that:

$$\beta \geq \frac{U_1 + \frac{1}{4}(U_1^+ - U_1)}{\frac{1}{2}U_1^+ + \frac{1}{2}U_1}$$

and so:

$$\begin{aligned} u_1\left(\sigma_1^{stop}, \sigma_2\right) &\geq \left(\alpha \cdot U_1^+ + (1-\alpha) \cdot U_1\right) \cdot \beta \\ &\geq \left(\frac{1}{2} \cdot U_1^+ + \frac{1}{2} \cdot U_1\right) \cdot \frac{U_1 + \frac{1}{4}(U_1^+ - U_1)}{\frac{1}{2}U_1^+ + \frac{1}{2}U_1} \\ &= U_1 + \frac{1}{4}(U_1^+ - U_1) \end{aligned}$$

However, $u_1\left(\sigma_1, \sigma_2\right) = U_1$. By the assumption that the utility functions are strictly competitive, $U_1^+ = U_1 + \epsilon'$ for some non-negligible function ϵ'. Thus:

$$u_1\left(\sigma_1^{stop}, \sigma_2\right) > u_1\left(\sigma_1, \sigma_2\right) + \epsilon$$

for a non-negligible function ϵ. We conclude that when $1 \geq \alpha \geq \frac{1}{2}$, P_1 prefers to change its strategy to σ_1^{stop}.

- *Case 2 – $0 \leq \alpha \leq \frac{1}{2}$:* This is almost equivalent to the previous case. We claim that in this case P_2 prefers to change its strategy to σ_2^{stop}. By Eq. (7) we have:

$$u_2\left(\sigma_1, \sigma_2^{stop}\right) = \left((1-\alpha) \cdot U_2^+ + \alpha \cdot U_2\right) \cdot \beta$$

Let $\alpha = \frac{1}{2} - \delta$, where $\delta \geq 0$. We write:

$$u_2\left(\sigma_1, \sigma_2^{stop}\right) = \left((1-\alpha) \cdot U_2^+ + \alpha \cdot U_2\right) \cdot \beta$$

$$
\begin{aligned}
&= \left(\left(\frac{1}{2}+\delta\right)\cdot U_2^+ + \left(\frac{1}{2}-\delta\right)\cdot U_2\right)\cdot\beta \\
&= \left(\delta\cdot\left(U_2^+ - U_2\right) + \frac{1}{2}\cdot U_2^+ + \frac{1}{2}\cdot U_2\right)\cdot\beta \\
&\geq \left(\frac{1}{2}\cdot U_2^+ + \frac{1}{2}\cdot U_2\right)\cdot\beta
\end{aligned}
$$

Using Eq. (8) we know that:

$$\beta \geq \frac{U_2 + \frac{1}{4}(U_2^+ - U_2)}{\frac{1}{2}U_2^+ + \frac{1}{2}U_2}$$

and thus:

$$
\begin{aligned}
u_2\left(\sigma_1, \sigma_2^{stop}\right) &\geq \left(\frac{1}{2}\cdot U_2^+ + \frac{1}{2}U_2\right)\cdot\beta \\
&\geq \left(\frac{1}{2}\cdot U_2^+ + \frac{1}{2}U_2\right)\cdot\frac{U_2 + \frac{1}{4}(U_2^+ - U_2)}{\frac{1}{2}U_2^+ + \frac{1}{2}U_2} \\
&= U_2 + \frac{1}{4}(U_2^+ - U_2)
\end{aligned}
$$

Recall again that $u_2(\sigma_1, \sigma_2) = U_2$ and $U_2^+ > U_2 + \epsilon'$ where ϵ' is non-negligible (by the assumption of strict competitiveness). We therefore conclude that:

$$u_2\left(\sigma_1, \sigma_2^{stop}\right) > u_2(\sigma_1, \sigma_2) + \epsilon$$

for some non-negligible function ϵ. Hence, when $\frac{1}{2} \geq \alpha \geq 0$, P_2 prefers to change its strategy to σ_2^{stop}.

We have shown that for every α, there is a party P_i whose utility is higher if it changes its strategy from σ_i to σ_i^{stop}. Hence, (σ_1, σ_2) is not a computational-Nash equilibrium, in contradiction. ■

This completes the proof of Theorem 4.6. ■

Remark: Note that when $\alpha = 1/2$, *both* parties prefer to change their strategy. In actuality, we can show that when β is almost 1, for almost every α, it is better for both parties to change their strategy to $\vec{\sigma}^{stop}$ (assuming that the other party follows the prescribed strategy $\vec{\sigma}$). Nevertheless, $\vec{\sigma}^{stop}$ is not a fair prescribed strategy for rational reconstruction because as we have shown, in such a case only *one* of them will learn the result (with very high probability). Thus, $\vec{\sigma}^{stop}$ does not result in o^{both}.

4.3 Impossibility with Arbitrary Auxiliary Input (Non-Simultaneous)

A closer look at the Kol and Naor mechanism (in the non-simultaneous model), raises another possible problem, namely the possibility that one of the parties has some prior information about the secret. The problem is that the party with the long list sees the entire secret s at the definitive iteration before it is supposed to broadcast it. Suppose that the long party has some prior information about the secret s, or has access to some "place" where it can check whether the possible

current secret s' is the real secret (e.g. the secret is a password to some web server, and the party can just type the password and sees if it works). In such a case, there is no incentive for the party to broadcast the secret at the definitive iteration, and therefore the party with the short list does not learn the secret.

In this section, we ask whether this weakness in the protocol is inherent (for the non-simultaneous model). In order to formalize this question, we assume that the parties have access to some membership oracle $\mathcal{O}$, and each party can query the oracle to ask whether the current possible secret s' is the actual secret. We assume that the parties are run in polynomial time and so can only query the oracle a polynomial number of times. Clearly, this implies that fair secret sharing with auxiliary input is only possible if the secret is taken from a super-polynomial domain. We model it this way because otherwise there is a non-negligible a priori probability of guessing the secret anyway.

We stress that this is just one type of auxiliary information. However, a proof of impossibility in the face of such an oracle suffices to demonstrate impossibility in the presence of *arbitrary auxiliary input*.

Defining membership-auxiliary information. We define a fair reconstruction mechanism with membership-auxiliary information, to be a mechanism that achieves fairness, even when the parties have an access to such an oracle. Formally, a membership Oracle $\mathcal{O} : \mathcal{S} \to \{0,1\}$ is defined as follows:

$$\mathcal{O}_s(x) = \begin{cases} 1 & x = s \\ 0 & \text{otherwise} \end{cases}$$

For simplicity, we assume that the oracle is never wrong. We define fair reconstruction mechanism with membership-auxiliary information as follows:

Definition 4.12 *Let P_1, P_2 be two parties, let $\mathcal{U}$ be a set of natural utility values for the parties (as in Definition 2.5), and let $\mathcal{O}$ be a membership oracle. We say that $(\Gamma^{\mathcal{O}}, \vec{\sigma}^{\mathcal{O}})$ is a* fair reconstruction mechanism with membership-auxiliary information *for the set $\mathcal{U}$, if the mechanism is fair (as in Definition 2.6), even when both parties have access to the oracle $\mathcal{O}$.*

We now show another impossibility result in the non-simultaneous model:

Theorem 4.13 *For any set of strictly competitive utility functions $\mathcal{U}$ (see Definition 4.5), there does not exist a fair reconstruction mechanism with membership-auxiliary information for the set $\mathcal{U}$, in the non-simultaneous model.*

Proof Sketch: The proof for this theorem is the same proof as Theorem 4.6, with some modifications. Actually, in Theorem 4.6 we assumed that the mechanism achieves correctness. However, our mechanism does not necessarily guarantee this property, and this requires some changes. Let $(\Gamma^{\mathcal{O}}, \vec{\sigma}^{\mathcal{O}})$ be the mechanism. We make the following modifications to the proof of Theorem 4.6:

- **Default output notation:** In the original proof, it is guaranteed that every default output of the parties is in the set $\{s, \perp\}$ except with negligible probability. This was crucial for the definition of the set Ω, and is actually a property of the correctness assumption. In our proof, $(\Gamma^{\mathcal{O}}, \vec{\sigma}^{\mathcal{O}})$ does not guarantee correctness, and therefore, the above does not hold. Therefore, we have to modify the definition of the default output: denote by a_i^{original} the output value of party P_1 on round i in the original protocol. We define our "new default output" as follows:

$$a_i = \begin{cases} a_i^{\text{original}} & \text{if } \mathcal{O}(a_i) = 1 \\ \perp & otherwise \end{cases}$$

That is, in order to compute the new default output at round i, P_1 calculates the default output of that round (as it did in the previous proof), and queries the oracle with that value. If the oracle returns 1, then P_1 outputs it (using the oracle answer, we know that this value is s). Otherwise, it sets the new default output to be $\perp$. We define the "new default output" for the party P_2 in the same way. With this modification, our new default outputs will also be in the set $\{s, \perp\}$, as needed in the proof. Given this modification, we use the same definition of good coins (Definition 4.7). The proof that its size is the same as before (except that requirement 2 holds with probability 1, in contrary to the original proof, where it may not hold with at most negligible probability).

- **The alternative strategy σ^{stop}:** We use exactly the same alternative strategy σ^{stop}, but emphasize that we use the new default output notations in the calculation of this strategy.
- **About U^f**: In the original proof, we assumed correctness, and therefore we assumed that the probability of causing the other party to output an incorrect value, and achieve U^f, is negligible. In other words, we assumed that:

$$\Pr\left[o_2^{false} \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] = 0 \qquad \text{and} \qquad \Pr\left[o_1^{false} \mid (\sigma_1, \sigma_2^{stop}), \rho \in \Omega_{p(k)}\right] = 0$$

The above was true because whenever $\rho \in \Omega_{p(k)}$, then for both parties the default outputs were always in $\{s, \perp\}$. This assumption was used in the calculations of Claim 4.11. For example, this is an implicit assumption in the following equation:

$$\begin{aligned} u_1\left(\sigma_1^{stop}, \sigma_2 \mid \rho \in \Omega_{p(k)}\right) &= \Pr\left[o_1^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^+ + \Pr\left[o^{\text{both}} \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1 \\ &\quad + \Pr\left[o_1^- \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^- + \Pr\left[o_2^+ \mid (\sigma_1^{stop}, \sigma_2), \rho \in \Omega_{p(k)}\right] \cdot U_1^{--} \end{aligned}$$

In the proof here, we do not assume correctness, and therefore, apparently, we cannot ignore U^f. However, we note that including the possibility of causing the other party to output a wrong secret can only make the expected utility even bigger (irrespective of the value of U^f and even if $U^f > U$). In order to see this, recall that we are considering the case that P_1 *always learns* the secret. Since U^f is the expected utility when P_1 *does not* learn the secret (but causes to the other party to output an incorrect value), it follows that U^f does not occur at all. Thus, the probability of obtaining U^f is zero. Note that there is still the possibility that P_1 *learns* the secret, and *also* causes to the other party to output an incorrect value. In this case, the utility that it achieves may be greater than or equal to U_1^+. However, this only makes the expected utility of following strategy σ_1^{stop} even greater, as required.

■

Auxiliary information – simultaneous channels. We remark that in the simultaneous channels model, it is possible to obtain fair reconstruction with auxiliary information. In particular, the protocols of [6, 12] achieve this.

5 Positive Results

5.1 Fully Independent Mechanisms for $n \geq 3$ with Simultaneous Channels

In this section we show that utility dependence is not always essential. In particular, we show that it is possible to construct a *utility independent* fair reconstruction mechanism for the case of t-out-of-n secret sharing, where $n \geq 3$. Our protocol is also resilient to coalitions, where the size of the coalition is less than $\lceil \frac{t}{2} \rceil$. Our protocol can use as a subprotocol any one of the protocols that were suggested in previous papers (for example [6, 12, 1, 11]). For the sake of concreteness, we use the protocol of Gordon-Katz [6] and show that our protocol inherits its properties. Specifically, we achieve a computational Nash equilibrium that survives iterated deletion of weakly dominated strategies. Using a subprotocol with a stronger Nash equilibrium will yield a stronger guarantee for our protocol as well.

5.1.1 Preliminaries

Utility functions for the multiparty case. Until this point in this paper we considered the two party case only. We now introduce the "standard" assumptions [7, 12] on the utility functions that are used in the multiparty case. Let r be the run of an execution, containing the random tapes of all parties and all messages sent, and let $o(r) = (o_1(r), \ldots, o_n(r))$ be the outcome of the execution r such that $o_i(r) = 1$ if and only if party P_i learned the secret in the run r. Let r and r' be two possible executions. The assumptions on the utilities are:

1. $u_i(r) = u_i(r')$ if $o(r) = o(r')$. Stated in words, the utilities of the parties depend only on the outcome of the run (that is, it depends only on who learned and who did not learn the secret).[2]

2. For every i, if $o_i(r) = 1$ and $o_i(r') = 0$, then $u_i(r) > u_i(r')$. That is, each party prefers to learn the secret than not learning it, irrespective of who else learns the secret.

3. If $o_i(r) = o_i(r')$ and $\mathsf{wt}(o(r)) < \mathsf{wt}(o(r'))$ (where $\mathsf{wt}(x)$ is the Hamming weight of the bit vector x), then $u_i(r) > u_i(r')$. (Each party prefers that as few parties as possible learn the secret.)

The third assumption is general enough to allow parties to have different utilities when different subsets of parties (of the same size) learn the secret.

Coalitions. We take the definition of an equilibrium that is resilient to coalitions from [1]. Let $\mathcal{C}$ denote a subset of the parties, or $\mathsf{coalition}$. Informally, a joint strategy $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ is *k-resilient* if for any coalition of parties $\mathcal{C}$ of size at most k that jointly deviate from $\vec{\sigma}$, none of the parties in the coalition can gain a higher utility than they would have gained by running $\vec{\sigma}$. Let $\mathcal{P} = \{1, \ldots, n\}$ be the set of indices of the parties and let Σ_i be the set of all possible strategies for party P_i (note that Σ_i may be an infinite set). For any subset of parties $\mathcal{C} \subset \mathcal{P}$, let $\Sigma_{\mathcal{C}}$ be the set of all possible joint strategies for parties in $\mathcal{C}$, and let $\sigma_{\mathcal{C}} \in \Sigma_{\mathcal{C}}$ be a joint strategy for those parties.

[2] We ignore here the issue of correctness and assume that it is always achieved. This is reasonable in the simultaneous channels model where all protocols – including ours – guarantee correctness except with negligible probability.

Definition 5.1 (k-resilient equilibrium) *For any non-empty subset of parties $\mathcal{C} \subseteq \mathcal{P}$, $\sigma_{\mathcal{C}} \in \Sigma_{\mathcal{C}}$ is a* group best response *for $\mathcal{C}$ to $\sigma_{-\mathcal{C}} \in \Sigma_{-\mathcal{C}}$ if for every strategy $\sigma'_{\mathcal{C}} \in \Sigma_{\mathcal{C}}$ and for every $i \in \mathcal{C}$, it holds that:*

$$u_i(\sigma_{\mathcal{C}}, \sigma_{-\mathcal{C}}) \geq u_i(\sigma'_{\mathcal{C}}, \sigma_{-\mathcal{C}})$$

*A join strategy $\vec{\sigma}$ is a k-*resilient Nash equilibrium *if for all $\mathcal{C} \subset \mathcal{P}$ for which $|\mathcal{C}| \leq k$, it holds that $\sigma_{\mathcal{C}}$ is a group best response for $\mathcal{C}$ to $\sigma_{-\mathcal{C}}$.*

Weakly dominated strategies and iterated deletion. We refer the reader to [7] for a formal definition of this notion; our protocol directly inherits this property from the protocol of [6] and we therefore omit technical details of this notion.

5.1.2 The Gordon-Katz Protocol

We now describe the protocol of [6] for rational secret sharing. This protocol is used as a building block in our protocol. The protocol of [6] uses an online dealer (or alternatively a protocol for secure computation), who in every round creates shares of the real secret with probability β, and shares of a fake secret with probability $1 - \beta$, where β is a parameter described below. All the shares are signed by the dealer. At every round, the parties reveal their shares (at the same time, using the *simultaneous* channel). If a party refuses to reveal its share, or if a party reveals an incorrect share (i.e. a share with an invalid signature), then all the parties stop the execution and the game is terminated. Thus, if a party decides to remain silent in any particular round and all other parties reveal their shares in that round, then with probability β the party can reconstruct the secret and be the only one to learn the secret (thereby gaining utility U^+). However, with probability $1 - \beta$, the party will learn nothing and will cause the execution to halt, with the result that it gains utility U^-. Thus, such a party takes a risk by not cooperating.

Consider now for simplicity the case of 2-out-of-2 secret sharing. Assume that party P_i defects in a given round, while the other party does not (i.e., the other party sends its share in the round). The expected utility of party P_i when it defects (quits, or remains silent in order to learn the secret alone) in the current round is:

$$u_i(\mathsf{defect}) = \beta \cdot U_i^+ + (1 - \beta) \cdot U_i^-$$

This is because with probability β the real secret is revealed in this round and with probability $1 - \beta$ a fake secret is revealed. Since P_i does not send its share, it is the only party to obtain the real or fake secret. Thus, with probability β it obtains utility U_i^+ (when the real secret is revealed) and with probability $1 - \beta$ it obtains utility U_i^- (observe that when a fake secret is revealed in this round, the execution is halted and neither party learns the secret).

In contrast, the expected utility of party P_i when it cooperates and broadcasts its share in every round (assuming that the other party plays according to the prescribed strategy) is U_i; that is $u_i(\mathsf{coop}) = U_i$. Thus, as long as $u_i(\mathsf{coop}) > u_i(\mathsf{defect})$, the prescribed strategy is a Nash equilibrium. Plugging in our calculations of these expected utility values, we have that the protocol achieves a Nash equilibrium as long as for every i it holds that

$$\beta \cdot U_i^+ + (1 - \beta) \cdot U_i^- < U_i.$$

By simple manipulation we obtain that this holds as long as for every i,

$$\beta < \frac{U_i - U_i^-}{U_i^+ - U_i^-}.$$

(We remark that this can be achieved as long as $U_i^+ \neq U_i^-$ because otherwise we divide by 0; this follows by the assumption in Eq. (1) (see Section 4.1). However, if the difference between U_i^+ and U_i^- is negligible, then the expected number of rounds of the protocol – which is $1/\beta$ – will not be polynomial. This can be solved by assuming strict competitiveness, or just that there exists a polynomial p such that for every i and all sufficiently large k's it holds that $U_i^+ > U_i^- + 1/p(k)$.)

Note that the protocol assumes that the parties can determine whether a given value is the real secret or a fake one. This can be achieved by simply adding a bit that indicates whether the value is the real secret or not. Gordon and Katz proved that for β set as above, the prescribed strategy of this protocol is a *Nash equilibrium that survives iterated deletions of weakly dominated strategies*. They also claim that in the case of t-out-of-n secret sharing their protocol is *k-resilient* for $k = t - 1$. See [6] for more details. We also refer to [11] regarding modifications necessary to obtain resilience to backward induction.

5.1.3 Our Protocol

We now present a protocol for t-out-of-n secret sharing for $n \geq 3$ and any $t \leq n$ that is completely independent of all the actual utility values. That is, our protocol achieves a Nash equilibrium that survives iterated deletions of weakly dominated strategies and is $(\lceil \frac{t}{2} \rceil - 1)$-resilient for *any* set of natural utility functions that are strictly competitive. This is in contrast to all of the protocols that were suggested in previous papers, and in contrast to our impossibility result in Section 4.1 for the two-party case.

The protocol idea. Our protocol is based on an observation made by [16] that extraneous shares help to achieve fairness. In particular, consider the case of t-out-of-n secret sharing where t is strictly less than n, and consider what happens when $t^* > t$ parties come together to reconstruct the secret using the naive reconstruction mechanism of just broadcasting the shares. In this case, if a single party deviates from the prescribed strategy and is silent, then it will not prevent the others from learning the secret. This is because there are still $t^* - 1 \geq t$ shares that are broadcast. This does not suffice because the strategy of being silent still dominates the strategy of broadcasting (nothing is lost by being silent). However, if any penalty is introduced that makes being silent "risky", then parties will have an incentive to cooperate. In particular, if instead of using the naive mechanism, the Gordon-Katz protocol is used with $\beta = 1/2$, then no party has an incentive to not cooperate. In order to see why this is the case, we analyze the expected utility of a party P_i who is silent in some round. If in that round the fake secret is revealed, then P_i will not learn the secret and will obtain the utility value U_i^-. In contrast, if the real secret is revealed then it *will* learn the secret. However, since there are $t^* - 1 \geq t$ other parties who do participate, then all parties still learn the secret and the utility gained is U_i (and *not* U_i^+). Thus, the expected utility of P_i is

$$\frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^- < U_i.$$

Thus, P_i can only lose by being silent. Note that this can be extended so that any coalition of $t^* - t$ parties has nothing to gain by being silent. The crucial point here is that the above mechanism

works *irrespective of the actual utility values of the parties*. However, it does not suffice for our goal because it requires that $t^* > t$ parties participate in the reconstruction which cannot always be guaranteed. In particular, in the highly important case of n-out-of-n secret sharing (which has many applications in secure computation for example) it is not possible to ever have $t^* > t$.

Our protocol works by achieving the effect as above even when $t^* = t$. As a first attempt, consider what happens if instead of sharing the secret using a t-out-of-n secret sharing scheme, a $\lceil \frac{t}{2} \rceil$-out-of-$n$ secret sharing scheme is used instead. In this case, when t parties come together to reconstruct the secret there are always $t - \lceil \frac{t}{2} \rceil$ extraneous shares (i.e., the effect is that of $t^* \approx 2t$), and so fairness can be achieved as above. Of course, this does not work because the scheme is no longer a t-out-of-n secret sharing scheme because only $\lceil \frac{t}{2} \rceil$ parties are needed to reconstruct instead of t. This problem can be solved by first sharing a random value r to mask the secret using a t-out-of-n secret sharing scheme, and then sharing $r \oplus s$ using a $\lceil \frac{t}{2} \rceil$-out-of-$n$ secret sharing scheme. First observe that now only t parties can reconstruct the secret, because $r \oplus s$ reveals nothing about s as long as r remains hidden. Now, in order to reconstruct, the parties first broadcast their shares of r naively; if any party is silent then they all abort and learn nothing. Then, after r is reconstructed, they use the Gordon-Katz protocol with $\beta = 1/2$ in order to reconstruct $r \oplus s$. The key observation is that in this second phase there are t parties, whereas only $\lceil \frac{t}{2} \rceil$ shares are needed. Thus, no coalition of size less than $\lceil \frac{t}{2} \rceil$ has any incentive to be silent (as shown above, being silent in this case only decreases their expected utility). As above, this holds irrespective of the actual utility values of the parties.

Our protocol:

The Dealer protocol: Let n be the number of parties, let t be the threshold and let s be the secret to be shared.

- Choose a random $r \in_R \{0,1\}^{|s|}$.
- Generate shares of r using a t-out-of-n secret sharing scheme; let $(r_1, ..., r_n)$ be the resulting shares.
- Generate shares of $r \oplus s$ using a $\lceil \frac{t}{2} \rceil$-out-of-$n$ secret sharing scheme; let $(p_1, ..., p_n)$ be the resulting shares.
- For every $i \in \{1, \ldots, n\}$, send the pair (r_i, p_i) to party P_i.

All the shares are signed by the dealer.

Reconstruction – the prescribed strategy σ_i for party P_i: Let σ_i^{GK} be the prescribed strategy for party P_i in the protocol of Gordon-Katz with $\beta = \frac{1}{2}$. The prescribed strategy in our protocol is as follows:

- Broadcast the share r_i.
- If all other parties that are participating in the reconstruction protocol broadcast their shares, and the shares are correct, then reconstruct r. Otherwise, abort.
- Run the protocol of Gordon-Katz with $\beta = 1/2$ using strategy σ_i^{GK}, using as input the shares $p_1, \ldots, p_n$ of $r \oplus s$.[3] Let w be the output received from this execution.

[3]The secure computation of Gordon-Katz can be defined so that in every iteration the parties input their shares

- If $w = \perp$, then output $\perp$. Otherwise, output $s = w \oplus r$.

Theorem 5.2 *Let $n \geq 3$ and $2 < t \leq n$ be integers. Then, the prescribed strategy $\vec{\sigma}$ of the game $(\Gamma, \vec{\sigma})$ is a Nash equilibrium that survives iterated deletions of weakly dominated strategies and is $(\lceil \frac{t}{2} \rceil - 1)$-resilient, for any set $\mathcal{U}$ of natural utility functions that are strictly competitive. Furthermore, the expected number of rounds of the game is $O(1)$.*

Proof: The expected number of phases of the Gordon-Katz protocol is $1/\beta$. Since we use $\beta = 1/2$, the expected number of phases is constant. When using an online dealer the number of rounds is essentially the same as the number of phases. However, when using secure computation, a constant-round protocol must be used; see [10]. We now proceed to show that $\vec{\sigma}$ is a Nash equilibrium that survives iterated deletions of weakly dominated strategies. (This involves considering only a single party deviating from the strategy; we will demonstrate resilience to coalitions afterwards.)

We first show that the expected utility of any deterministic aborting strategy is lower than the expected utility of the prescribed strategy. As is usual, we assume that the only effective actions a party can take are to send the correct share and be silent. This is because sending an incorrect share has the same effect as being silent which is to cause the execution to halt. (We ignore the negligible probability that a signature can be successfully forged.) Clearly, if any party is silent in the first phase where r is reconstructed, then all parties obtain U^-. Thus, any non-zero probability of following this strategy always yields a lower utility. From here on, we consider the utility of being silent in the second phase of the protocol. Denote by abort_i^j the deterministic strategy where party P_i follows the prescribed strategy in rounds 1 through $j-1$, and then is silent in round j. If the protocol concludes before round j, then all parties learn the secret and obtain U. Otherwise, with probability $\beta = 1/2$ no parties learn the secret (in the case that a fake secret is revealed in this round) and with probability $1 - \beta = 1/2$ all parties still learn the secret (because $t - 1$ shares are sent and only $\lceil \frac{t}{2} \rceil$ shares are actually needed). Denoting by $\mathsf{halt}(< j)$ the event that the protocol halts before round j, we have that:

$$\begin{aligned}
u_i(\mathsf{abort}_i^j, \sigma_{-i}) &= \Pr[\mathsf{halt}(<j)] \cdot U_i + \Pr[\neg\mathsf{halt}(<j)] \cdot \left(\frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^-\right) \\
&= \Pr[\mathsf{halt}(<j)] \cdot U_i + (1 - \Pr[\mathsf{halt}(<j)]) \cdot \left(\frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^-\right) \\
&= \frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^- + \Pr[\mathsf{halt}(<j)] \cdot \left(U_i - \frac{1}{2} \cdot U_i - \frac{1}{2} \cdot U_i^-\right) \\
&= \frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^- + \Pr[\mathsf{halt}(<j)] \cdot \left(\frac{1}{2} \cdot U_i - \frac{1}{2} \cdot U_i^-\right) \\
&< \frac{1}{2} \cdot U_i + \frac{1}{2} \cdot U_i^- + \left(\frac{1}{2} \cdot U_i - \frac{1}{2} \cdot U_i^-\right) \\
&= U_i
\end{aligned}$$

where the inequality is due to the fact that $\Pr[\mathsf{halt}(< j)] < 1$. Thus, for every $i \in \{1, \ldots, n\}$ and for every j,

$$u_i(\vec{\sigma}) > u_i(\mathsf{abort}_i^j, \sigma_{-i})$$

$p_1, \ldots, p_n$. The functionality checks the validity of the shares and then generates either fake shares (with probability $1 - \beta$) or new real shares of s, as reconstructed from $p_1, \ldots, p_n$.

In order to consider probabilistic strategies, we note that the view of any party until the protocol terminates is independent of the secret (it contains just fake random secrets). Thus, any strategy of being silent during the execution can be written as a strategy that assigns an a priori probability that the party is silent in every round. Specifically, a probabilistic strategy is a set of probabilities $\mathcal{P}_i = \{p_i^j\}_{j=1}^{\infty}$ such that the probability that the party P_i follows strategy abort_i^j equals p_i^j and $\sum_{j=1}^{\infty} p_i^j = 1$. Based on this, we have that for every such strategy $\mathcal{P}_i$:

$$u_i(\mathcal{P}_i, \sigma_{-i}) = \sum_{j=1}^{\infty} p_i^j \cdot u_i(\mathsf{abort}_i^j, \sigma_{-i}) < \sum_{j=1}^{\infty} p_i^j \cdot u_i(\vec{\sigma}) = u_i(\vec{\sigma}) \cdot \sum_{j=1}^{\infty} p_i^j = u_i(\vec{\sigma}) = U_i$$

where the inequality is from our proof above that for every i and j, $u_i(\vec{\sigma}) > u_i(\mathsf{abort}_i^j, \vec{\sigma}_{-i})$. This shows that $\vec{\sigma}$ is a Nash equilibrium. The fact that it survives iterated deletion of weakly dominated strategies is *identical* to the proof that the Gordon-Katz mechanism has this property. We therefore do not repeat the proof and refer the reader to [6].

It remains to show that our protocol is $(\lceil \frac{t}{2} \rceil - 1)$-resilient. This is shown via a reduction to the fact that the protocol of Gordon-Katz is $(t'-1)$-resilient when t' is the threshold used. (Note that we use threshold $t' = \lceil \frac{t}{2} \rceil$ and thus this is consistent.) Assume by contradiction that our protocol is not $(\lceil \frac{t}{2} \rceil - 1)$-resilient. This implies that there exists a coalition of parties $\mathcal{C}$ of size less than $\lceil \frac{t}{2} \rceil$ and a strategy $\sigma'_{\mathcal{C}}$ such that for some $i \in \mathcal{C}$,

$$u_i(\sigma'_{\mathcal{C}}, \sigma_{-\mathcal{C}}) > u_i(\sigma_{\mathcal{C}}, \sigma_{-\mathcal{C}}).$$

Consider now the strategy $\sigma'_{\mathcal{C}}$ in the first phase of our protocol. If $\sigma'_{\mathcal{C}}$ instructs any of the parties to not broadcast their share of r in this phase, then as we have seen, this can only lower the utility. Thus, if this is the case, then there exists another strategy $\sigma''_{\mathcal{C}}$ in which all parties in $\mathcal{C}$ broadcast their share of r with probability 1 and for every $i \in \mathcal{C}$, $u_i(\sigma''_{\mathcal{C}}, \sigma_{-\mathcal{C}}) \geq u_i(\sigma'_{\mathcal{C}}, \sigma_{-\mathcal{C}})$, which is greater than $u_i(\vec{\sigma})$ by the assumption. Now, since $\sigma''_{\mathcal{C}}$ instructs all parties to broadcast their share of r with probability 1, they always proceed to the second phase. Let $\hat{\sigma}_{\mathcal{C}}$ be the strategy of $\sigma''_{\mathcal{C}}$ in the second phase of the protocol. Clearly, if $u_i(\hat{\sigma}_{\mathcal{C}}, \sigma_{-\mathcal{C}}^{GK}) \leq u_i(\sigma_{\mathcal{C}}^{GK}, \sigma_{-\mathcal{C}}^{GK})$, then the expected utility of every $i \in \mathcal{C}$ when running our protocol with $\sigma''_{\mathcal{C}}$ would be less than or equal to the expected utility of every $i \in \mathcal{C}$ when running our protocol with the prescribed strategy $\sigma_{\mathcal{C}}$. Thus, it must hold that for some $i \in \mathcal{C}$,

$$u_i(\hat{\sigma}_{\mathcal{C}}, \sigma_{-\mathcal{C}}^{GK}) > u_i(\sigma_{\mathcal{C}}^{GK}, \sigma_{-\mathcal{C}}^{GK}).$$

However, the above now relates to the utility purely in the Gordon-Katz protocol. Since this protocol is resilient to any coalitions of size less than the threshold, it follows that it is resilient to any coalition of size $\lceil \frac{t}{2} \rceil - 1$ or less (because the threshold used in the secret sharing of $r \oplus s$ is $\lceil \frac{t}{2} \rceil$). Since by the assumption $|\mathcal{C}| < \lceil \frac{t}{2} \rceil$, we have a contradiction. We conclude that our protocol is $(\lceil \frac{t}{2} \rceil - 1)$-resilient, as required. ■

Remark – the case of $t = 2$: Observe that Theorem 5.2 is stated only for the case that $t > 2$. (It is actually possible to state it also for $t = 2$; however, in this case the theorem is vacuous because when $t = 2$ the resilience is to coalitions of size $\lceil \frac{t}{2} \rceil - 1 = 0$.) The reason for this is that the impossibility results for utility-independent two-party rational secret sharing hold when $t = 2$ irrespective of the value of n (this is because two parties alone must be able to reconstruct fairly). We remark that if $t = 2$, $n \geq 3$ and one is willing to assume that reconstruction always takes place with $t^* > t$ parties, then utility independence *is* possible, as shown by [16]. In this case, the solution

is to just run the second phase of the reconstruction protocol above directly with s (instead of with $r \oplus s$).

5.1.4 Optimality of our Protocol with Respect to Strong Coalitions

We now show that it is impossible to achieve *utility-independent* fair reconstruction with strong coalitions of size $\lceil \frac{t}{2} \rceil$ or greater by showing that this would imply fair reconstruction for the case of $n = 2$. That is, we show that it is impossible to have a fair reconstruction protocol with a k-resilient equilibrium for every $k \geq \lceil \frac{t}{2} \rceil$. As a result, this shows that it is impossible to achieve a so-called *strong resilient equilibrium* as defined by Abraham et al. [1] (a strategy $\vec{\sigma}$ is strongly resilient if it is k-resilient for all $k \leq t^* - 1$, where t^* is the number of parties participating in the reconstruction phase). We leave open the question of whether or not it is possible to achieve fair reconstruction in the presence of "rational coalitions"; i.e., coalitions where the parties within the coalition are also rational [3].

Theorem 5.3 *Let $n \in \mathbb{N}$, let $t \leq n$ and let $k = \lceil \frac{t}{2} \rceil$. Then, there does not exist a fair reconstruction that is k-resilient and U^+-independent, even in the simultaneous channels model.*

Proof: The proof of this theorem follows by a simple reduction to the two-party case. Assume by contradiction that there exists such a mechanism, and let $\vec{\sigma}$ be the prescribed strategy. We construct a two-party mechanism for parties $\hat{P}_1$ and $\hat{P}_2$ as follows: party $\hat{P}_1$ runs the strategy of parties $P_1, \ldots, P_{\lceil \frac{t}{2} \rceil}$ in $\vec{\sigma}$ and $\hat{P}_2$ runs the strategy of parties $P_{\lceil \frac{t}{2} \rceil + 1}, \ldots, P_n$ in $\vec{\sigma}$. This means that $\hat{P}_1$ internally emulates the execution of all parties $P_1, \ldots, P_{\lceil \frac{t}{2} \rceil}$; all messages that are sent between these parties are dealt with internally by $\hat{P}_1$ and all messages that are sent to the other parties are sent to $\hat{P}_2$ (with a clear labeling that states which message is intended for which party). The simple observation is that both $\hat{P}_1$ and $\hat{P}_2$ constitute coalitions of size at most $\lceil \frac{t}{2} \rceil$ in the mechanism. Thus, by the assumption, the reconstruction must be fair, and so $\vec{\sigma}$ (as transformed to the two-party setting) is a Nash equilibrium for $\hat{P}_1$ and $\hat{P}_2$. However, since in $\vec{\sigma}$ all parties $P_1, \ldots, P_n$ receive the secret, both $\hat{P}_1$ and $\hat{P}_2$ also receive the secret. Thus, we obtain a U^+-independent fair reconstruction mechanism for the two-party case, in contradiction to Theorem 4.4. ■

5.2 U^f-Dependent Correct Reconstruction in the Non-Simultaneous Model

In this section, we address the basic question of whether or not it is possible to construct a *fair and correct* reconstruction mechanism using non-simultaneous channels even if $U_i^f \geq U_i$ (recall that the mechanism of [12] achieves correctness when $U_i^f < U_i$). We answer this in the positive by constructing a mechanism that works as long as the value of U_i^f for each party P_i is known (in the same way that the values of U_i^+, U_i, U_i^- and U_i^{--} are known).

The idea behind the mechanism. We will consider the two party case only; the extension to the multiparty case is straightforward. We assume familiarity with the protocol of Kol and Naor [12]; see the beginning of Section 4.2 for a short description of the protocol and why it does not guarantee correctness; this will be used below. Looking closely at the strategy for breaking correctness in the Kol-Naor mechanism, it arises because the first party to send its list element in an iteration has no way of verifying if the current round is the definitive round or not. This is necessary because if the long party (i.e., the party with the long list) can determine that the current round is the

definitive one before sending its element, then it can simply not send its element with the result being that it learns the secret without the other party learning it. Despite this, our key observation is that it is not necessary that all of the fake iterations be the same, as in the Kol-Naor mechanism. Rather, we introduce additional rounds with the property that the first party in each such round *knows* that the round is fake while the second party does not. Now, if a second party halts in such a round under the premise that it reached the end of the list, then the first party will know that it has cheated and so will abort. The result is that the parties do not learn the secret and so the parties both gain utility U^-, and the second party does not gain U^f. By adding enough of these additional rounds, we have that the probability that a party successfully achieves U^f is low enough so that a higher expected utility is obtained by playing $\vec{\sigma}$ and obtaining U.

In more detail, in our new mechanism we give the first party to speak in a given iteration a way to distinguish between *types* of rounds. While in the original protocol, there were two types of rounds (one real round and many fake rounds), we have three different types of rounds:

- **The real round**: that is, the definitive iteration. In this round, the secret is exposed.
- **Fake rounds**: The goal of these fake rounds is to decrease the probability of achieving U^+. The number of fake rounds is chosen according to a geometric distribution with parameter β that depends on U^+, U, U^-, as defined in [12]. The number of fake rounds is exactly the same as in the Kol-Naor mechanism [12].
- **Completely fake rounds**: The goal of these rounds is to decrease the probability of achieving U^f. In contrast to fake rounds, at every *completely fake round* the first party will be informed that the round is *completely fake*. That is, it is given a boolean vector that indicates for every round whether it is *completely fake* or whether it is either fake or real (of course, without distinction regarding the latter). The number of completely fake rounds is chosen according to a geometric distribution with a parameter α (we will show how to choose α below).

The main idea is as follows. The first party at every iteration cannot know which is the definitive round because it does not know which of the non-completely fake rounds is the real one. This is exactly the same as in the Kol-Naor mechanism. Furthermore, the second party cannot stop early and fool the first party into accepting an incorrect value because it does not know which of the rounds are completely fake and which are fake. Observe that if the second party stops on a completely fake round then the first party detects this and aborts. In this case, the second party will obtain U^- instead of U or U^f, and so certainly loses.

The protocol itself: Let $Dealer(s, \beta)$ be the dealer in the Kol-Naor mechanism. In our new protocol, we use the value of the utility U^f in order to achieve correctness. Our new Dealer takes the shares generated by $Dealer(s, \beta)$ and extends the list for the parties.

The Dealer Protocol:

- **Input**: A secret s and a set of utility functions of the parties $\mathcal{U} = \{(U_i^+, U_i, U_i^-, U_i^f)_{i\in\{1,2\}}\}$[4].
- **Output**: A list of shares for each of the parties, denoted L_{short}^{out} and L_{long}^{out}. Moreover, each party receives a boolean list, denoted $\mathsf{blist}_{\text{long}}$, $\mathsf{blist}_{\text{short}}$. The values of these lists are false (informing the party that this round is *completely fake*), or $\mathsf{unknown}$ (meaning that this

[4]We ignore U_i^{--} in order to be consistent with the original protocol. However, the protocol can easily be extended to deal with U_i^{--}.

round may be either the *real round* or a *fake round*). Note that when a party is the second party to broadcast in a round its value is always unknown.

- **The protocol:**

  1. Compute β as defined in [12], call $Dealer(s, \beta)$ and obtain the lists L^{in}_{short} and L^{in}_{long}. Let ℓ_{short} be the length of the list L^{in}_{short}, and ℓ_{long} the length of L^{in}_{long}. Initialize $\mathsf{blist}_{\text{long}}$, $\mathsf{blist}_{\text{short}}$, L^{out}_{long} and L^{out}_{short} to be empty lists ($\emptyset$).
  2. Set $i = 0$ and compute α as described below.
  3. Repeat ℓ_{long} times:
     - Toss a coin with probability α to be 1, and with probability $1 - \alpha$ to be 0. Let b be the result of the current toss.
     - While $b = 0$:
       * Add a *completely fake round*. That is, choose a possible secret (from $\mathcal{S}$),[5] and append it to the lists L^{out}_{long} and L^{out}_{short}. Moreover, create at random a permutation of broadcast order: toss a random coin $\mathsf{order} \in_R \{long, short\}$:
         · If $\mathsf{order} = long$ then the first party to talk in this round will be the long party. Thus, append to $\mathsf{blist}_{\text{long}}$ the boolean false and append to $\mathsf{blist}_{\text{short}}$ the value unknown.
         · If $\mathsf{order} = short$ then the first party to talk in this round will be the short party. Thus, append to $\mathsf{blist}_{\text{short}}$ the value false, and append to $\mathsf{blist}_{\text{long}}$ the value unknown.
       * Toss the coin again.
     - When $b = 1$, add *a real round or a fake round*: Take the ith element from $L^{in}_{short}, L^{in}_{long}$ and place it in L^{out}_{short}, L^{out}_{long}. (If there is no such element in L^{in}_{short} since it has already ended, then take the element from L^{in}_{long} and put it in L^{out}_{long} only). Append to the $\mathsf{blist}_{\text{long}}$ the boolean unknown. If $i \leq \ell_{\text{short}}$ (L^{in}_{short} has not ended yet), append also to the short party's boolean list the value unknown.
  4. Assign the shares to P_1, P_2 as $Dealer(s, \beta)$, and send the long party the list $\mathsf{blist}_{\text{long}}$ and the short party the list $\mathsf{blist}_{\text{short}}$.

Before proceeding, we remark that the expected number of rounds of our protocol is $1/\alpha$ times the expected number of rounds in the original Kol-Naor mechanism. Moreover, the vector blist does not affect the equilibrium demonstrated by Kol and Naor because the number of "fake rounds" equals the number of rounds in their protocol.

Let $\sigma^{KN} = (\sigma^{KN}_1, \sigma^{KN}_2)$ be the prescribed strategy in the Kol-Naor mechanism. The new prescribed strategy is $\vec{\sigma} = (\sigma_1, \sigma_2)$ as follows:

- **Party P_i:**
  - Follow the exact same strategy σ^{KN}_i.
  - When σ^{KN}_i instructs to leave the game at round i and output s', check that $\mathsf{blist}[i] \neq \mathsf{false}$. If yes, output s'. Otherwise, output $\perp$.

[5] Note that at each *completely fake round* we construct a round which looks exactly like an original round in the Kol-Naor mechanism. That is, we give each party a masked secret, authentication information, and other details as described in the protocol. We omit these details as they are not important for our main point.

Setting α: Note that party P_i will obtain U_i^f if it quits at round j (where j is *not* the definitive iteration), it is the second party in that round, and $\mathsf{blist}[i] = \mathsf{unknown}$ (recall that if $\mathsf{blist}[i] = \mathsf{false}$ then P_i obtains U_i^-). Let σ_i' be the strategy in which party P_i quits at the non-definitive round j. The expected utility of this strategy is:

$$u_i(\sigma_i', \sigma_{-i}) = \alpha \cdot U_i^f + (1-\alpha) \cdot U_i^-$$

In contrast, the expected utility of party P_i playing according to the prescribed strategy σ_i under the assumption that the other parties follow σ_{-i} is U_i. Thus, in order to guarantee that $u_i(\sigma_1, \sigma_2) > u_i(\sigma_i', \sigma_{-i})$ we require that:

$$\begin{aligned} U_i &> \alpha \cdot U_i^f + (1-\alpha) \cdot U_i^- \\ &= \alpha \cdot U_i^f + U_i^- - \alpha \cdot U_i^- \\ &= U_i^- + \alpha \cdot \left(U_i^f - U_i^- \right) \end{aligned}$$

and so

$$\alpha \cdot \left(U_i^f - U_i^- \right) < U_i - U_i^-.$$

We conclude that $u_i(\sigma_1, \sigma_2) > u_i(\sigma_i', \sigma_{-i})$ if and only if

$$\alpha < \frac{U_i - U_i^-}{U_i^f - U_i^-}.$$

This should hold for every $i \in \{1, 2\}$. Combining the above with the analysis of [12] we obtain that the mechanism has the same equilibrium as in the mechanism of Kol-Naor. The expected running time of the Kol-Naor mechanism is $O(\frac{2}{\beta})$, and its expected share size is $O\left(\frac{2\log 2}{\beta}(\log \frac{1}{\beta} + \log \frac{U_{max}}{\epsilon})\right)$, where U_{max} is an upper bound on the payoffs that the party may receive. Thus, the expected running time of our new protocol is $O\left(\frac{2}{\alpha \cdot \beta}\right)$, and the expected share size is $O\left(\frac{2\log 2}{\alpha \cdot \beta}(\log \frac{1}{\beta} + \log \frac{U_{max}}{\epsilon})\right)$.

Acknowledgements

We thank Ran Canetti for asking the question of whether it is possible to achieve fair secret sharing *with auxiliary input* with non-simultaneous channels (see Section 4.3).

References

[1] I. Abraham, D. Dolev, R. Gonen and J.Y. Halpern. Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. In the 25*th PODC*, pages 53–62, 2006.

[2] I. Abraham, D. Dolev and J.Y. Halpern. Lower Bounds on Implementing Robust and Resilient Mediators. In the 5*th TCC*, Springer-Verlag (LNCS 4948), pages 302–319, 2008.

[3] B.D. Bernheim, B. Peleg and M.D. Whinston. Coalition-Proof Nash Equilibria. In *Journal of Economic Theory*, Vol. 42, pages 1–12, 1987.

[4] R. Cleve. Limits on the Security of Coin Flips when Half the Processors are Faulty. In 18*th STOC,* pages 364–369, 1986.

[5] Y. Dodis and T. Rabin. Cryptography and Game Theory. In *Algorithmic Game Theory*, Cambridge University Press, 2007.

[6] S.D. Gordon and J. Katz. Rational Secret Sharing, Revisited. In the 5*th Conference on Security and Cryptography for Networks (SCN)*, pages 229–241, 2006.

[7] J. Halpern and V. Teague. Rational Secret Sharing and Multiparty Computation. In the 36*th STOC*, pages 623–632, 2004.

[8] G. Fuchsbauer, J. Katz, E. Levieil and D. Naccache. Efficient Rational Secret Sharing in the Standard Communication Model. *Cryptology ePrint Archive,* Report #2008/488, 2008.

[9] J. Katz. Bridging Game Theory and Cryptography: Recent Results and Future Directions. In 5*th TCC*, Springer-Verlag (LNCS 4948), pages 251–272, 2008.

[10] J. Katz, R. Ostrovsky and A. Smith. Round Efficiency of Multi-party Computation with a Dishonest Majority. In *EUROCRYPT 2003*, Springer-Verlag (LNCS 2656), pages 578–595, 2003.

[11] G. Kol and M. Naor. Cryptography and Game Theory: Designing Protocols for Exchanging Information. In the 5*th TCC*, Springer-Verlag (LNCS 4948), pages 320–339, 2008.

[12] G. Kol and M. Naor. Games for Exchanging Information. In the 40*th STOC*, pages 423–432, 2008.

[13] S. Izmalkov, S. Micali, and M. Lepinski. Rational Secure Computation and Ideal Mechanism Design. In the 46*th FOCS*, pages 585–595, 2005.

[14] M. Lepinski, S. Micali, C. Peikert, and A. Shelat. Completely Fair SFE and Coalition-Safe Cheap Talk. In the 23*rd PODC*, pages 1–10, 2004.

[15] M. Lepinski, S. Micali, and A. Shelat. Collusion-Free Protocols. In the 37*th STOC*, pages 543–552, 2005.

[16] A. Lysyanskaya and N. Triandopoulos. Rationality and Adversarial Behavior in Multiparty Computation. In *CRYPTO 2006*, Springer-Verlag (LNCS 4117), pages 180–197, 2006.

[17] A. Shamir. How to Share a Secret. In *Communications of the ACM*, 22(11):612–613, 1979.

[18] S.J. Ong, D. Parkes, A. Rosen and S. Vadhan. Fairness with an Honest Minority and a Rational Majority. In the 6*th TCC*, Springer-Verlag (LNCS 5444), pages 36–53, 2009.

A Modeling Utility Functions

The standard definition of a utility function is a function that maps the outcome of an execution of the protocol to the parties "satisfaction" or benefit from this outcome. The standard definition states that the utilities depend only on the outcome of the execution of the mechanism. If we use

this definition, then the question of whether there exists a single mechanism that is independent of the utility functions of the parties is easy, if we allow asymptotics. Specifically, if we view the utility functions as constant, then we can use the mechanisms that were suggested in the previous papers (like the mechanism of [6, 12] etc.) and set β to equal $1/k$ where k is the security parameter. This will then guarantee that the mechanism will achieve equilibrium *for all large enough values of* k. (This is because for all large enough values of k, β will be smaller than the defined ratio between the utility functions.)

Although this may seem reasonable at first sight, we argue that it is highly unsatisfactory. In particular, although we happily work with asymptotics when it comes to cryptographic hardness assumptions, when it comes to actually using a scheme we need to set a concrete security parameter. This is not done by mere guessing but by making careful calculations based on the best known algorithms for solving the hard problem being used. The important point is that this state of art is public knowledge and so can be used to calculate the security parameter.[6] In contrast, it is not at all clear that it is possible to bound the utility of parties. In particular, a personal gain of a given party in a game may be based on their investment portfolio and political connections. This information is often not public knowledge and so cannot be used to reasonably estimate β.

Due to the above, we propose that utility functions should be modeled as functions that are *polynomial in the security parameter*. All known protocols work for this modeling, and we believe that this is what was actually implicitly assumed.

[6] Of course, it may be the case that a better algorithm is known but has not been published. However, by taking conservative values of the security parameter, we can make a reasonable estimate of the needed value.