Utility Dependence in Correct and Fair Rational Secret Sharing*

Gilad Asharov and Yehuda Lindell

Department of Computer Science
Bar-Ilan University, Israel
gilad_asharov@yahoo.com, lindell@cs.biu.ac.il

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 basic setting, rational secret sharing cannot be achieved without it. On the positive side, we show that by somewhat relaxing the standard assumptions on the utility functions, it is possible to achieve utility independence. In addition to the above, 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 is not at all unlikely.) 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.

1 Introduction

Recently, there has been much interest in the intersection between cryptography and game theory [6, 5, 10, 3, 1, 9, 10]. 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

* Research supported by the Israel Science Foundation (grant No. 781/07).

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 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 [6]. 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 essentially 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 to learn the secret than to not learn 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 meaning that no party can see what the others broadcast before sending its own). Since simultaneous channels are problematic to implement in practice, a recent breakthrough was made that achieved rational secret sharing in non-simultaneous channels [10]. However, the protocol of [10] (and a follow-up protocol by [7]) 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 [10] 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?*

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 to learn than not learn, 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 *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 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 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 [2] 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 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 actually 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. [5, 10]) are U^f-independent. This leaves open the question regarding U^f-independence with non-simultaneous channels. We prove that:

Theorem 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 that states that a U^f-independent mechanism guarantees correctness. We then prove that in the non-simultaneous channels model, it is not possible to construct a *correct* reconstruction mechanism.

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

1. We present a two-party reconstruction mechanism for rational secret sharing that works in the non-simultaneous model. This mechanism uses the actual values of U^f; given the impossibility result of Theorem 3, this is inherent.
2. We present a *multiparty* reconstruction mechanism that uses simultaneous channels and is *independent of all utility values*, under a relaxation of the learning-preference assumptions. Namely, we assume that a party prefers to be the only one to learn the secret but once one other party has learned the secret it makes no difference if all learn it. In fact, it suffices to assume that even though each party prefers that as few other parties as possible learn the secret, the utility if all but 1 or all but 2 parties learn is the same (i.e., it makes no difference if *all parties* learn the secret or if *almost all parties* learn the secret).

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

Related work. The question of rational secret sharing was first introduced by [6]. They showed that there does not exist a mechanism with a constant number of rounds, with a 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 [5, 1, 9, 10], and for the non-simultaneous model in [10, 7]. The basic question that we ask regarding utility independence was proposed in [6]. The first partial answer to this question was given by [1] 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 [14] who gave a partial answer in their model. Among other things, we have shown that it is *not* possible for the important case of 2-out-of-2 secret sharing. The works of [13, 11] 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 [16, 14, 1].

2 Definitions and Preliminaries

We denote by $\mathcal{S}$ an efficiently samplable distribution for choosing the secret to be shared, by SHARE the secret sharing scheme and by $(\Gamma, \boldsymbol{\sigma})$ the reconstruction mechanism. Definitions of secret sharing and Nash equilibria can be found in the full version.

Outcome and utilities. The outcome of an execution of a game Γ with some strategy profile $\boldsymbol{\sigma}$ 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 paper 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.

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

In this paper, 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 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. 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 [6, 5, 10], we assume that each party always prefers to learn the secret than to not learn 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 [10] 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 4 *Let $\mathcal{U} = \{(U_i^+, U_i, U_i^-, U_i^{--}, U_i^f)_{i\in\{1,2\}}\}$ 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 $k \in \mathsf{N}$ it holds that $U_i^+(k) \geq U_i(k) \geq U_i^-(k) \geq U_i^{--}(k) \geq 0$ and $U_i^f(k) \geq U_i^-(k)$.*

We remark that in all previous works, it was formally assumed that $U_i^-(k) = U_i^{--}(k)$, even though none of the protocols utilized this fact. We have not defined it in this way because 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. We note that all previous protocols can be modified to work with the value U_i^{--}. We also note that our

lower bounds hold even if $U_i^- = U_i^{--}$, and so we do not assume anything about the value U_i^{--}.

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 $\epsilon(\cdot)$ [10, 8]. 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 [3, 8] (although our results hold even if local computation by each party is unbounded). We define computationally fair reconstruction mechanisms in this light:

Definition 5 *Let $\mathcal{U}$ be a set of natural utility functions for P_1 and P_2 (as in Definition 4). We say that a mechanism $(\Gamma, \boldsymbol{\sigma})$ is a* fair reconstruction mechanism for $\mathcal{U}$ *if $\boldsymbol{\sigma}$ is a computational Nash Equilibrium and if the probability that the result is not o^{both} when both parties follow $\boldsymbol{\sigma}$ 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 6 *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 $(\Gamma, \boldsymbol{\sigma})$ 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 $(\Gamma, \boldsymbol{\sigma})$ 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 implies the properties of complete fairness and correctness in the presence of adversarial behavior.[1] 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

[1] We consider the two-party case only because we only deal with the case of no coalitions in this paper, and in the case of no coalitions we have an honest majority and so fairness and correctness (in the presence of a malicious adversary) can be achieved. This case is therefore not interesting.

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). We remark that we will move freely between protocols in a cryptographic setting with an adversary $\mathcal{A}$ and mechanisms involving rational adversaries playing a game in order to achieve utility. This is due to the fact that a mechanism trivially defines a protocol and vice versa. 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.

We first 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 by $\mathcal{S}$ 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 7 *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 the 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 the 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 $(\Gamma, \boldsymbol{\sigma})$ 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:

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

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

3.2 U^+-Independence vs Fairness and U^f-Independence vs Correctness

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 8 *If there exists a U^+-independent fair reconstruction mechanism for a 2-out-of-2 secret sharing scheme (as in Definition 6), then there exists a completely fair reconstruction protocol (as in Definition 7) for the scheme.*

Proof: Let $(\Gamma,\boldsymbol{\sigma})$ 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 $(\Gamma,\boldsymbol{\sigma})$ as described above. 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:

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

Let $\sigma^{\mathcal{A}}$ be the corresponding behavioral strategy of the adversary $\mathcal{A}$ in the game Γ. Note that 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 $\boldsymbol{\sigma}$, 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 $\boldsymbol{\sigma}$ 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, $u_i(\sigma_i, \sigma_{-i}) = U_i$. Thus, there exists a non negligible function ϵ' (even if U_i is negligible), such that:

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

in contradiction to the assumption that $\boldsymbol{\sigma}$ is a computational Nash equilibrium for Γ. We therefore conclude that the protocol π induced from $(\Gamma, \boldsymbol{\sigma})$ is completely fair, as in Definition 7. ■

U^f-independence implies correctness. The following is proved analogously to Proposition 8:

Proposition 9 *If a fair reconstruction mechanism for a 2-out-of-2 secret sharing scheme is U^f-independent (as in Definition 6), then it achieve correctness (as in Definition 7).*

4 Negative Results

4.1 Impossibility for U^+-Independence

As we have mentioned, Proposition 8 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 [2]. (Specifically, secure computation can be used to generate shares of a random bit, which are then reconstructed using the mechanism. By Proposition 8, 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.

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:

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

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 10 *Let $(\Gamma, \boldsymbol{\sigma})$ 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 $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$, 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 Sketch: We start with some 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. 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 11 *Let $\mathcal{U}$ be a set of natural utility functions for P_1 and P_2 (as in Definition 4), and let the mechanism $(\Gamma, \boldsymbol{\sigma})$ be a* fair reconstruction mechanism for $\mathcal{U}$ *(as in Definition 5). For every round $i \geq 0$, the following must hold:*

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

Proof Sketch: Assume by contradiction that the above does not hold. Without loss of generality, assume that there exists an i such that

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

In the proof, we consider an alternative strategy σ_1^i for P_1 which is identical to the prescribed strategy σ_1 except that it instructs the party P_1 to quit before broadcasting the message in round i. Assuming that the other party (P_2) does broadcast its share in that round, and that the execution reaches round i, we have that P_1 outputs a_{i+1} while P_2 outputs b_i. Using the contradicting assumption, it follows that:

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

That is, with probability at least $2\beta_1$ the outcome is o_1^+, and therefore P_1 gains U_1^+ while P_2 gains only U_2^{--}. Thus, the expected utility of P_1 is at least

$$2\beta_1 \cdot U_1^+ + (1 - 2\beta_1) \cdot U_1^{--} = 2\beta_1(U_1^+ - U_1^{--}) + U_1^{--} = 2U_1 - 2U_1^{--} + U_1^{--} > U_1$$

where the last equality is by the assumption that U_i is non-negligibly greater than U^- and U_i^{--} (note that if $U_i \approx U^-$ then P_i has no reason to play at all). Thus, the strategy σ_1^i of stopping in round i is a better strategy for P_1, in contradiction to the assumption that $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$ is a Nash equilibrium.

We stress that some important details are omitted from this proof sketch. For example, it does not take into account the probability that round i is actually reached in the execution or the possibility of negligible failure; see the full version for details. ■

We use the above claim to prove our lower bound. Now, consider the case that the secret is a uniformly distributed k-bit string. In such a case, the probability that any party outputs the correct secret before receiving any message is negligible (i.e., $\Pr[a_0 = s] = \Pr[b_0 = s] = \mu(k)$ for some negligible function μ). By simple induction, we have that for every i:

$$\Pr[a_i = s] \leq 2i\beta + \mu(k) \qquad \text{and} \qquad \Pr[b_i = s] \leq 2i\beta + \mu(k)$$

and so

$$\sum_{i=1}^{2r(k)} \Pr[a_i = s] \leq \sum_{i=1}^{2r(k)} 2i\beta + \mu(k) \approx 4\beta \cdot r^2(k)$$

where $r(k)$ denotes the expected number of rounds; i.e., $E[R^{\Gamma}_{(\sigma_1,\sigma_2)}] = r(k)$. By Markov, $\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} \geq 2r(k)\right] \leq \frac{1}{2}$ and so $\Pr\left[R^{\Gamma}_{(\sigma_1,\sigma_2)} < 2r(k)\right] > \frac{1}{2}$. Now, if $R^{\Gamma}_{(\sigma_1,\sigma_2)} < 2r(k)$ then for some $i \in \{1,\ldots,2r(k)\}$ it holds that $a_i = s$ (because at the end, both parties must output s). Thus,

$$\sum_{i=1}^{2r(k)} \Pr[a_i = s] > \frac{1}{2}.$$

We conclude that

$$\frac{1}{2} < \sum_{i=1}^{2r(k)} \Pr[a_i = s] \leq 4\beta r^2(k)$$

implying that $r(k) > \frac{1}{\sqrt{8\beta}}$. (Note that the theorem bounds $r(k) > \frac{1}{8\sqrt{\beta}}$ and not what we have shown here. This is due to additional factors that we have omitted from this sketch; see the full version for details.) ■

Using Theorem 10 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 $r(k) < \frac{1}{8\sqrt{\beta}}$. Specifically, take $U_i^+ \geq 64r^2(k) \cdot \left(U_i - U_i^{--}\right) + U_i^{--}$. This suffices because in such a case

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

and thus $r(k) \leq \frac{1}{8\sqrt{\beta_i}}$ in contradiction.

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 [5, 9]. 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 $\boldsymbol{\sigma}$ of [10, 7] 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 [10] 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 $\mathcal{U}$ fulfill the assumptions in Definition 4. 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 [10] for details.

As pointed out in [10, 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 [7]. As we have mentioned, [10] 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; see [10]. We are now ready to formally state the theorem.

Theorem 12 *There do not exist U^f-independent fair reconstruction mechanisms for strictly competitive utility functions in the non-simultaneous model.*

By Proposition 9, 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 7.

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 $\boldsymbol{\sigma}$ 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 σ^{stop} both of the parties will output the correct s in the round that they quit. Next, we prove that when both parties follow $\boldsymbol{\sigma}^{stop}$, with high probability one of them learns the secret while the other does not. We conclude by showing that the prescribed strategy $\boldsymbol{\sigma}$ is not a computational Nash equilibrium by showing that one of the $\boldsymbol{\sigma}^{stop}$ strategies has a better expected utility than $\boldsymbol{\sigma}$. 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'. The proof of this appears in the full version.

5 Positive Results

5.1 U^f-Dependent 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 $U_i^f \geq U_i$. We answer this in the positive by constructing a mechanism that works as long as it knows the value of U_i^f for each party P_i (in the same way that the mechanism knows the values of U_i^+, U_i, U_i^- and U_i^{--}).

The idea behind the mechanism. We will consider the two party case only, but the idea works for the multiparty case as well. We assume familiarity with the protocol of Kol and Naor [10]; 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 could check if the current round is the definitive one before sending its element, it could learn 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 second party in each such round knows that the round is fake while the first party does not. Now, if a first party prematurely aborts on such a round, then it will gain only U^-, and not U^f (because the second party knows that the first party has cheated and just aborts outputting $\perp$). 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 $\boldsymbol{\sigma}$ and obtaining U. See the full version for a detailed description and proof.

5.2 Full Independence for $n \geq 3$ with Relaxed Assumptions

In this section we show that utility dependence is not always essential. In particular, we show that for a certain reasonable relaxation of the utility functions, it is possible to construct a *utility independent* fair reconstruction mechanism for the case of t-out-of-n secret sharing, where $n \geq 3$. We do not claim that our assumptions always hold or should be used; rather our aim here is to show that utility independence can sometimes be achieved.

The "standard" assumptions [6, 10] typically used for the utility functions are that a party always prefers to learn than not. Furthermore, assuming that a party learns, the fewer others that learn the better. We relax these assumptions, and assume that each party prefers to learn the secret alone, but once *one* of the other parties learns the secret it doesn't matter how many other parties learn it (thus U_i^+ denotes the utility when it alone learns, and U_i denotes the utility that it learns along with any positive number of other parties).

In addition to the above, we assume that the utility functions are polynomial in the security parameter, and that there is a non negligible difference between them. That is, there exists a polynomial $p(\cdot)$, such that for infinitely many k's it holds that: $U_i \geq U_i^- + \frac{1}{p(k)}$. (Our impossibility result for U^+-independence when $n = 2$ holds for such utility functions.) This is a natural extension of the strict differences between the utility functions, as defined in [10], when they are modeled as functions of the security parameter. (We remark that U_i^+ may equal U_i; we only need a non-negligible difference between U_i and U_i^-.) Note that when the above does not hold, it means that P_i's utility when not learning is essentially the same as when learning. Thus, P_i may as well not participate at all and this case is not interesting. Our protocol assumes simultaneous channels in order to achieve U^f-independence. As we showed in Theorem 12, it is impossible to achieve U^f independence with non-simultaneous channels.[2]

[2] A version of our protocol for the non-simultaneous model can be constructed using the techniques of [10] and our protocol in Section 5.1. However, note that the protocol

The protocol idea. The idea behind our protocol is to enable one of the parties to learn the secret even when the others do not. Now, once this party has learned the secret, it is not possible for any other party to obtain U^+. Thus, the other parties can either continue with the execution of the protocol and obtain U, or they can quit and obtain only U^- (which is strictly less than U by the assumption that $U_i \geq U_i^- + \frac{1}{p(k)}$). The main question is how to construct a protocol so that one of the parties can learn the secret, but only after there are $t^* \geq t$ parties participating in the reconstruction phase, but then enable the residual $t-1$ parties to reconstruct the secret without the cooperation of the party who already learned the secret.

We achieve this in the following way. Let s be the secret to be shared; for simplicity assume that $s \in \{0,1\}^k$ (where k is the security parameter). The dealer chooses a random $r \in_R \{0,1\}^k$ and generates shares of r and s with threshold t and shares of $r \oplus s$ with threshold $t-1$ (overall three sets of shares). The dealer then sends each party its shares. Before proceeding we note that no set of $t-1$ parties can reconstruct the secret s, because even though a set of this size can learn $r \oplus s$, without knowing r this is of no help. In addition, ignoring issues of rationality and utility, it is possible for every set of t parties to obtain s by just reconstructing the shares of s, or by reconstructing r and $r \oplus s$ (where the latter requires only $t-1$ to participate).

We now informally describe our reconstruction protocol. In the first phase of the protocol $t^* \geq t$ parties reconstruct r by simply sending their shares to all others. In the second phase, the t^* parties reconstruct s by sending their shares one at a time consecutively (here it is crucial that a *simultaneous* channel *not* be used and so we use the simultaneous channel as a non-simultaneous one, by having every party wait until it receives all previous messages before sending its own). Note that at the end of the second phase, the last $t^* - t + 1$ parties can reconstruct the secret alone, and thus, they may not send their shares. If any of the parties does not send their share in the first phase, or if any of the first $t-1$ parties does not send their share in the second phase, then all parties abort and output $\perp$. At the end of this phase, unless all have aborted, there remain $t-1$ parties who have not learned the secret. These parties continue to the third phase. The crucial observation is that none of these parties can obtain U^+ since there are already $t^* - t + 1 \geq 1$ parties who have learned the secret s. We utilize this fact to use any one of the known rational reconstruction protocols while setting $\beta = \frac{1}{2}$ (where β is a parameter that usually depends on the utility values, like our β in the lower bound); observe that we fix β irrespective of the actual utility values. This works because at this point, once one party has learned the secret, the maximum possible utility the parties can obtain is U. In particular, even if only one party of the remaining $t-1$ parties learns the secret, its utility is still U because one party already knows the secret. Now, the known rational secret sharing protocols with $\beta = \frac{1}{2}$ all have the property that if the parties follow $\boldsymbol{\sigma}$ then they will obtain U (with probability 1). However, if they

---

for the non-simultaneous model needs to know the values of U_i^f, U_i and U_i^{--}, and therefore the result is only U^+-independent.

do not, then with probability $1-\beta$ they will obtain U^-. Thus, the expected utility by not following $\boldsymbol{\sigma}$ is $\frac{1}{2} \cdot U^- + \frac{1}{2} \cdot U < U$, and so the parties follow $\boldsymbol{\sigma}$ and all learn the secret.

Remark. Our construction can be extended to deal with the case that parties *do* prefer that as few as possible other parties learn, but *do not care* whether $t-2$ or $t-1$ parties learn (i.e., it does not make any difference if all learn, or all but one learn). This is a much milder relaxation on the utility functions; see the full version for details.

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. R. Cleve. Limits on the Security of Coin Flips when Half the Processors are Faulty. In 18*th STOC,* pages 364–369, 1986.
3. Y. Dodis and T. Rabin. Cryptography and Game Theory. In *Algorithmic Game Theory*, Cambridge University Press, 2007.
4. O. Goldreich. *Foundations of Cryptography: Volume 2 – Basic Applications*. Cambridge University Press, 2004.
5. 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.
6. J. Halpern and V. Teague. Rational Secret Sharing and Multiparty Computation. In the 36*th STOC*, pages 623–632, 2004.
7. 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.
8. J. Katz. Bridging Game Theory and Cryptography: Recent Results and Future Directions. In 5*th TCC*, Springer-Verlag (LNCS 4948), pages 251–272, 2008.
9. 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.
10. G. Kol and M. Naor. Games for Exchanging Information. In the 40*th STOC*, pages 423–432, 2008.
11. S. Izmalkov, S. Micali, and M. Lepinski. Rational Secure Computation and Ideal Mechanism Design. In the 46*th FOCS*, pages 585–595, 2005.
12. 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.
13. M. Lepinski, S. Micali, and A. Shelat. Collusion-Free Protocols. In the 37*th STOC*, pages 543–552, 2005.
14. A. Lysyanskaya and N. Triandopoulos. Rationality and Adversarial Behavior in Multiparty Computation. In *CRYPTO 2006*, Springer-Verlag (LNCS 4117), pages 180–197, 2006.
15. A. Shamir. How to Share a Secret. In *Communications of the ACM*, 22(11):612–613, 1979.
16. 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.