Rational Secret Sharing, Revisited

S. Dov Gordon* Jonathan Katz*†

Abstract

We consider the problem of secret sharing among n rational players. This problem was introduced by Halpern and Teague (STOC 2004), who claim that a solution is *impossible* for $n = 2$ but show a solution for the case $n \geq 3$. Contrary to their claim, we show a protocol for rational secret sharing among $n = 2$ players; our protocol extends to the case $n \geq 3$, where it is simpler than the Halpern-Teague solution and also offers a number of other advantages.

We also show how to avoid the continual involvement of the dealer, in either our own protocol or that of Halpern and Teague.

Our techniques extend to the case of rational players trying to securely compute an arbitrary function, under certain assumptions on the utilities of the players.

1 Introduction

The classical problem of *t-out-of-n secret sharing* [12, 1] involves a "dealer" D who wishes to entrust a secret s to a group of n players $P_1, \ldots, P_n$ so that (1) any group of t or more players can reconstruct the secret without further intervention of the dealer, yet (2) any group of fewer than t players has no information about the secret. As an example, consider the scheme due to Shamir [12]: assume the secret s lies in a field $\mathbb{F}$, with $|\mathbb{F}| > n$. The dealer chooses a random polynomial $f(x)$ of degree at most $t-1$ subject to the constraint $f(0) = s$, and gives the "share" $f(i)$ to player P_i (for $i = 1, \ldots, n$). Any set of t players can recover $f(x)$ (and hence s) by interpolation; furthermore, no set of fewer than t players has any information about s.

The implicit assumption above is that at least t players are willing to cooperate and pool their shares[1] when it is time to recover the secret; equivalently, at least t players are *honest* and hence at most $n - t$ players are *malicious*. Halpern and Teague [10] consider a scenario in which no players are (completely) honest, but instead all that is guaranteed is that at least t players are *rational* (as before, up to $n - t$ players may refuse to participate altogether). Depending on the utility functions of the players, Shamir's protocol may no longer succeed in this scenario [10]. Specifically, assume that all players prefer to learn the secret above all else, but otherwise prefer that the fewest number of other parties learn the secret. (We will treat the utilities of the players more carefully later in the paper.) Given these utility functions, no player has any incentive to reveal their share. Consider P_1: if strictly fewer than $t-1$ other players reveal their shares, then no one learns the secret regardless of whether P_1 reveals his share or not. If more than $t-1$ players reveal their shares, then everyone learns the secret and P_1's actions again have no effect. On the other hand,

*Department of Computer Science, University of Maryland. {gordon, jkatz}@cs.umd.edu

†Research supported by NSF Trusted Computing grants #0310499 and #0310751; NSF CAREER award #0447075; and US-Israel Binational Science Foundation grant #2004240.

[1] We assume adversarial behavior is limited to refusal to cooperate, and ignore the case that a player reports an incorrect share. In our context, such behavior is easily handled by having the dealer sign the shares.

if *exactly* $t-1$ other players reveal their shares, then P_1 learns the secret (using his share) but P_1 can prevent other players from learning the secret by *not* publicly revealing his share.

We can thus conclude the following about the game-theoretic equilibria of the above situation (definitions of Nash equilibrium and weakly dominating strategies are given in Section 2):

- For any t, n and any number t^* of rational, participating players, it is a Nash equilibrium for no one to reveal their shares.
- If $t < n$ and $t^* > t$, then it is a Nash equilibrium for all t^* participating players to reveal their shares. Note, however, that it is a weakly dominating strategy for each player *not* to reveal his share; thus, the Nash equilibrium likely to be reached is the one mentioned earlier in which no one reveals their share.
- If $t^* = t$, then having all t^* participating players reveal their shares is not even a Nash equilibrium. (Note that if $t = n$ then this will always be the situation.)

Thus, Shamir's protocol with the trivial reconstruction phase does not suffice in the presence of rational players. Does there exist *any* protocol for reconstructing the secret in which it is in rational players' best interests to follow the protocol? By generalizing the argument outlined earlier, Halpern and Teague rule out any protocol terminating in a *fixed* number of rounds. (Essentially, the above argument is applied to the last round and then backwards induction is used.) This leaves open the possibility of *probabilistic* protocols without a fixed upper bound on their round complexity, and indeed Halpern and Teague show the existence of such a protocol for the case of $n \geq 3$ parties. In contrast, they claim that a solution is *impossible* for $n = 2$ even if probabilistic protocols are allowed.

1.1 Our Results

We revisit the question of rational secret sharing, in the model of Halpern and Teague [10]. As perhaps our most surprising result, we show a simple, probabilistic protocol for $n = 2$ parties to reconstruct a shared secret, thus disproving the claim of Halpern and Teague mentioned earlier. Interestingly, the *proof* given by Halpern and Teague appears to be correct; the problem is that their *assumptions* about the types of protocols that might be used are too restrictive. By relaxing their assumptions in a reasonable way, we are able to circumvent their impossibility result.

Our protocol generalizes in a straightforward way to the case of $n \geq 3$ and arbitrary t. Although Halpern and Teague also claim a general solution of this sort, our solution is much simpler. Furthermore, for $n > 3$ our solution has a number of advantages as compared with the solution offered by Halpern and Teague; perhaps most importantly, our solution eliminates a second (undesirable) equilibrium that is present in the Halpern-Teague protocol. Other advantages of our approach are summarized in Section 3.3.

Both the Halpern-Teague protocol and our protocol (as initially described) require the continual, periodic involvement of the dealer. At best, this is inconvenient; at worst, this calls into question the motivation for the problem in the first place. We show in Section 4 an intuitively simple way to avoid the involvement of the dealer which applies in all scenarios considered here.

Our techniques extend to the more general case of rational players trying to securely compute an arbitrary function of their inputs, under certain assumptions on the utilities of the players. See Section 5 for further details.

1.2 Related Work

There has been much interest of late in bridging cryptography (in which guarantees are provided in the face of worst-case adversarial behavior) and game theory (which concerns itself only with rational deviations). Besides the work of Halpern and Teague, the most relevant prior work is the recent sequence of papers by Lepinski, et al. [7, 8] and Izmalkov, et al. [6]. We briefly compare their work to ours.

Lepinski, Micali, Peikert, and Shelat [7] show a protocol for *completely fair secure function evaluation* (SFE), in which all players receive output if any player receives output, even if up to $n-1$ players are malicious. In "standard" communication networks this is known to be impossible [2], and therefore Lepinski, et al. rely on the physical assumption of "secure envelopes" (see the discussion in [7] for the exact properties these should satisfy) to achieve their result. They then suggest how to use any protocol for completely fair SFE to implement *cheap talk* even in the presence of malicious coalitions; basically, this enables the players to reach a correlated equilibrium without having to rely on any external trusted party.

The work of Lepinski, Micali, and Shelat [8] and Izmalkov, Micali, and Lepinski [6] deals (directly or indirectly) with mechanisms for preventing coalitions in the first place. More specifically, these works are concerned with eliminating covert (e.g., steganographic) channels in the secure computation protocol itself so as to prevent signaling between players. Again, they achieve this by relying on physical assumptions (secure envelopes and, in the case of [6], ballot boxes) in addition to standard communication channels. A consequence of the work of Izmalkov, et al. (indeed, the main motivation for their work) is a protocol Π for securely implementing any mediated game Γ such that (informally) any equilibrium in Γ corresponds to an equilibrium in Π, and vice versa.

Comparison to our work. The work of Lepinski, et al. [7] as well as that of Izmalkov, et al. [6] both offer different solutions to the problem we consider here. The main difference in our work is that we focus on giving a very simple and efficient protocol for a very specific problem (and with a specific — but natural — set of utility functions), rather than (as in the case of [7, 6]) giving an inefficient solution for the general case. We are also interested mainly in analyzing single-player deviations (i.e., various forms of Nash equilibria), whereas the work of Izmalkov, et al. handles much more general classes of equilibria that include, in particular, those involving coalitions. (We remark, however, that our basic secret sharing protocol is resilient to coalitions as well as long as out-of-band communication is assumed not to occur.)

An additional difference between our work and that of [7, 8, 6] is that instead of relying on "secure envelopes," our solution relies on standard communication channels with the exception that we assume some form of *simultaneous broadcast* where each party broadcasts a message at the same time. (Equivalently, we do not allow "rushing.") Whether one finds this assumption realistic or not, we note that it is a strictly *weaker* assumption than secure envelopes since the former can be constructed from the latter but not vice versa.

Interestingly, our techniques for solving the specific problem of secret sharing extend to the case of secure computation of general functions. We thus obtain a conceptually simple protocol for completely fair SFE *under the assumption that all players are rational, and prefer to learn the outcome of the computation.* (The solution of [7] works even if players are arbitrarily malicious.) See Section 5 for further details.

2 The Model

We briefly review the model of Halpern and Teague, filling in some details they omit. As discussed earlier, we have a dealer D holding a secret s, and n players $P_1, \ldots, P_n$. There is also a threshold $t \leq n$, known to all players, which is fixed at the outset. A protocol proceeds in a sequence of *iterations*, where each iteration may consist of multiple *communication rounds*. At the beginning of each iteration, D distributes some information (privately) to each of the n players; at this point, no subset of fewer than t players should have any information about s. During an iteration, the dealer does not take part in the protocol. Instead, some set of t players, all of whom are assumed to be rational, run the protocol amongst themselves by simultaneously broadcasting messages in a series of rounds. (Halpern and Teague additionally allow private communication between the players but we do not need this.) For simplicity, we assume the same t players run the protocol in every iteration. At the end of an iteration, the protocol either terminates or proceeds to the next iteration. We assume the dealer is honest, and follows the protocol as specified. To rule out trivial protocols, we require that if t players follow the protocol in each iteration, then the secret is eventually reconstructed (with probability 1).

We stress that broadcast in a given round is assumed to occur simultaneously for all players; that is, we do not assume "rushing" as in standard multi-party computation protocols. In fact, rational secret sharing is easily seen to be impossible if rushing is allowed: all players will simply wait to see what other players do, and no one will ever broadcast anything.

In the above description, as in [10], the dealer is assumed to be involved at the beginning of each iteration. In Section 4, we show that it is possible for the dealer to be involved only once at the beginning of the protocol.

We let σ_i denote the (possibly randomized) strategy employed by player P_i, and let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ denote the vector of players' strategies. Following standard game-theoretic notation, we let $(\sigma_i', \vec{\sigma}_{-i}) \stackrel{\text{def}}{=} (\sigma_1, \ldots, \sigma_{i-1}, \sigma_i', \sigma_{i+1}, \ldots, \sigma_n)$; that is, $(\sigma_i', \vec{\sigma}_{-i})$ denotes the strategy vector $\vec{\sigma}$ with P_i's strategy changed to σ_i'.

Let $\mu_i(o)$ denote the utility of player P_i for the outcome o. For a particular outcome o of the protocol, we let $\delta_i(o)$ be a bit denoting whether or not P_i learns the secret, and let $\mathsf{num}(o) = \sum_i \delta_i(o)$; i.e., $\mathsf{num}(o)$ is simply the number of players who learn the secret. Following [10], we make the following assumptions about the utility functions of the players:

- $\delta_i(o) > \delta_i(o') \Rightarrow \mu_i(o) > \mu_i(o')$.
- Assuming $\delta_i(o) = \delta_i(o')$,
  $\mathsf{num}(o) < \mathsf{num}(o') \Rightarrow \mu_i(o) > \mu_i(o')$.

That is, players first prefer outcomes in which they learn the secret; as long as δ_i remains constant, players prefer strategies in which the fewest number of other players learn the secret.[2] We let $U_i(\vec{\sigma})$ denote the expected value of the utility of P_i under strategy vector $\vec{\sigma}$, and assume that rational players wish to maximize this value.

Our notion of a *protocol* corresponds to a *game* along with a prescribed strategy vector $\vec{\sigma}$. As in [10], we are interested in protocols whose prescribed strategy vector $\vec{\sigma}$ corresponds to a Nash equilibrium that survives iterated deletion of weakly dominated strategies. We review these definitions briefly, and refer the reader to [11, 10] for more extensive discussion.

[2] Halpern and Teague actually allow more complex utility functions than the ones described here. Our protocol handles such utility functions as well, but for simplicity we use the utility functions described here.

Definition 1 A vector of strategies $\vec{\sigma}$ is a *Nash equilibrium* if the following holds for all i: for any $\sigma_i' \neq \sigma_i$, we have $U_i(\sigma_i', \vec{\sigma}_{-i}) \leq U_i(\vec{\sigma})$. ♢

That is, given that all other players are following $\vec{\sigma}_{-i}$, there is no incentive for P_i to deviate and follow any strategy other than σ_i.

In general, multiple Nash equilibria may exist. An inherently "unstable" Nash equilibrium (i.e., one which is unlikely to be reached) is one in which any of the players' strategies are *weakly dominated* by other strategies. Informally, a strategy σ_i of player P_i is weakly dominated by another strategy σ_i' if (1) P_i is sometimes better off playing σ_i' than playing σ_i, and (2) P_i is never worse off playing σ_i' than playing σ_i. Recalling the example from the introduction, say a secret is shared using t-out-of-n secret sharing (with $t < n$) and consider the strategy vector in which all n players reveal their shares. This is a Nash equilibrium: the secret is reconstructed even if any single player deviates. On the other hand, for each player P_i, revealing the share is weakly dominated by *not* revealing the share: if less than $t-1$ or more than $t-1$ other players reveal their shares, then nothing changes; while if exactly $t-1$ other player reveal their shares then P_i learns the secret but no one else does.

Formal definitions follow.

Definition 2 Let S_i denote a set of strategies for P_i, and $S_{-i} \stackrel{\text{def}}{=} S_1 \times \cdots S_{i-1} \times S_{i+1} \cdots \times S_n$. A strategy $\sigma_i \in S_i$ is *weakly dominated* by a strategy $\sigma_i' \in S_i$ with respect to S_{-i} if (1) there exists a $\vec{\sigma}_{-i} \in S_{-i}$ such that $U_i(\sigma_i, \vec{\sigma}_{-i}) < U_i(\sigma_i', \vec{\sigma}_{-i})$ and (2) for all $\vec{\sigma}_{-i} \in S_{-i}$, it holds that $U_i(\sigma_i, \vec{\sigma}_{-i}) \leq U_i(\sigma_i', \vec{\sigma}_{-i})$.

Strategy σ_i is weakly dominated with respect to $S_1 \times \cdots S_n$ if there exists a $\sigma_i' \in S_i$ such that σ_i is weakly dominated by σ_i' with respect to S_{-i}. ♢

Definition 3 Let $\mathsf{DOM}_i(S_1 \times \cdots \times S_n)$ denote the set of strategies in S_i that are weakly dominated with respect to S_{-i}. Let S_i^0 denote the initial set of allowable strategies of P_i. For all $k \geq 1$, define S_i^k inductively as $S_i^k \stackrel{\text{def}}{=} S_i^{k-1} \setminus \mathsf{DOM}_i(S_1^{k-1} \times \cdots \times S_n^{k-1})$. Let $S_i^\infty \stackrel{\text{def}}{=} \cap_k S_i^k$.

We say that σ_i *survives iterated deletion of weakly dominated strategies* if $\sigma_i \in S_i^\infty$. ♢

3 Protocols for Rational Secret Sharing

3.1 The Halpern-Teague Solution

We provide a high-level overview of the solution of Halpern and Teague for 3-out-of-3 secret sharing. We later discuss how they propose to generalize their solution for $n > 3$ and $t \geq 3$.

The Halpern-Teague protocol proceeds as follows: at the beginning of each iteration, the dealer runs a fresh invocation of the Shamir secret-sharing scheme and sends the appropriate shares to each player. During an iteration, each player P_i flips a biased coin c_i which is equal to 1 with some probability α. The players then run what is essentially a secure multi-party computation protocol to compute the value $p = \bigoplus c_i$. In particular, this means that it is impossible to cheat (except for aborting the protocol; see below), or to learn information about the $\{c_i\}$ values of the other parties. If $p = c_i = 1$, player P_i broadcasts his share. If all shares are revealed, the secret is reconstructed and the protocol ends. If $p = 1$ and 0 or 2 shares are revealed, or if the secure computation of p was aborted, then all players refuse to run the protocol from then on (and so, effectively, the protocol is terminated). In any other case, players proceed to the next iteration.

Note that the secret is only reconstructed if $c_1 = c_2 = c_3 = 1$. Thus, assuming players act honestly, the expected number of iterations until the protocol terminates is α^{-3}.

To see intuitively why the above gives a Nash equilibrium, assume P_1, P_2 follow the protocol and consider whether P_3 should deviate. First note that there is no incentive for P_3 to bias c_3 to be 0 with higher probability, since when $c_3 = 0$ at least one of P_1, P_2 will not broadcast their shares in that iteration no matter what. There is also no incentive for P_3 to bias c_3 to be 1 with higher probability, either (although the secret may be reconstructed sooner, this has no effect on P_3's utility). It is also easy to see that, given $p = 0$ or $c_3 = 0$, there is no incentive for P_3 to deviate. Finally, when $p = c_3 = 1$, player P_3 does not know whether $c_1 = c_2 = 1$ (which occurs with probability $\frac{\alpha^2}{\alpha^2+(1-\alpha)^2}$) or $c_1 = c_2 = 0$ (which occurs with the remaining probability). Thus, if P_3 does not broadcast its share it runs the risk of having the protocol terminate without ever learning the secret. If α is set appropriately based on P_3's utility function, it is not in P_3's best interest to deviate.

For $n > 3$ and $t \geq 3$, Halpern and Teague suggest the following: the $t^* \geq t$ players split into 3 groups, with one leader designated per group. All players send their shares to their designated group leader, and the leaders then run the 3-out-of-3 solution given above. When these leaders are supposed to broadcast their share, they broadcast the shares of every member of their group to all players. The case of $t = 2$ requires some additional complications that we do not describe.

3.2 Our Solution

Recall that Halpern and Teague claim that rational secret sharing is *impossible* when $n = 2$. They implicitly assume that the dealer is limited to sending valid Shamir shares of the secret to the players at the beginning of each iteration, and they therefore focus only on what happens *during* an iteration. This assumption underlies the Halpern-Teague proof of impossibility for $n = 2$. However, we see no reason to impose this restriction on the dealer's actions. As we show in this section, removing this assumption makes a protocol possible even when $n = 2$, and drastically simplifies things for the case of general t, n.

Specifically, consider the following protocol: say the dealer holds a secret s which lies in a *strict subset* S of a field $\mathbb{F}$ (if s lies in some field $\mathbb{F}'$, this is easy to achieve by taking a larger field $\mathbb{F}$ containing $\mathbb{F}'$ as a subfield). We assume players know S. At the beginning of each iteration, with probability β (for some appropriate choice of β) the dealer generates a random Shamir sharing of s, and with probability $1-\beta$ the dealer generates a random Shamir sharing of an arbitrary element $\hat{s} \in \mathbb{F} \setminus S$. These shares are distributed to the players. Note that no player can tell whether they were given a share of $\hat{s}$ or the true secret s.

During an iteration, the players simply broadcast their shares. If in any iteration some player does not broadcast his share, the other players all refuse to reveal their shares in all subsequent iterations (and, again, the protocol is essentially terminated). Otherwise, all shares were broadcast and the players can reconstruct some value s'. If $s' \in S$ then the players know that this is the true secret, and can terminate the protocol successfully. If $s' \in \mathbb{F} \setminus S$, the players know this is an invalid secret and proceed to the next iteration.

Theorem 1 *For appropriate choice of β, the above protocol constitutes a Nash equilibrium for t-out-of-n secret sharing that survives iterated deletion of weakly dominated strategies.*

Proof We first consider the case of $t = n = 2$, and then discuss how to generalize the proof for arbitrary t, n. It is not hard to see that the protocol is a Nash equilibrium for appropriate choice of β: Say P_1 acts according to the protocol and consider whether P_2 has any incentive to deviate. Note that we need only consider strategies in which P_2 deviates in a single iteration (i.e., *single-stage deviations*) [3]. (In any infinitely-repeated game, there exists a deviation preferred by P_2 if and

only if there exists a single-stage deviation preferred by P_2 [3].) Without loss of generality, consider a deviation in the first iteration. The only possible deviation is for P_2 to refuse to broadcast his share. In this case, he learns the secret (while P_1 does not) with probability β, but with probability $1-\beta$ he will never learn the secret.

By re-scaling if necessary, we may assume that P_2's utility is 1 if he does not learn the secret, $U > 1$ if both P_2 and P_1 learn the secret, and $U^* > U$ if he learns the secret but P_1 does not. If P_2 follows the protocol, his utility is U. If P_2 deviates, his expected utility is $\beta \cdot U^* + (1-\beta)$. So as long as

$$\beta \cdot U^* + (1-\beta) < U,$$

it is in P_2's best interest to follow the protocol. For appropriate $\beta > 0$, then, the strategy profile in which both parties follow the protocol is a Nash equilibrium.

It is trivial to see that the same analysis holds for general t, n.

We next prove that our protocol survives iterated deletion of weakly dominated strategies by showing that *no* strategies are weakly dominated. We will again begin with the case $t = n = 2$. We show that for all deterministic strategies σ, σ' of P_1, there exist strategies τ, τ' of P_2 such that $U_1(\sigma, \tau) > U_1(\sigma', \tau)$ but $U_1(\sigma, \tau') < U_1(\sigma', \tau')$. This proves that all deterministic strategies of P_1 are incomparable, and so none are ever deleted (and extends immediately to show the same result for randomized strategies as well).

Let $h_i(\sigma, \tau)$ denote the history of actions (by both players) through iteration i given the indicated strategies σ and τ, with $h_0(\sigma, \tau)$ denoting the empty (starting) history. Let $A_i(\sigma, \tau)$ denote the action taken by P_1 in iteration i, again for the indicated strategies. We say a player *cooperates* in some iteration if they reveal their share, and *defects* if they do not.

Now take arbitrary deterministic strategies $\sigma \neq \sigma'$ for P_1. Let τ^0 be a strategy of P_2 and $i \geq 1$ be an integer such that

$$h_{i-1}(\sigma, \tau^0) = h_{i-1}(\sigma', \tau^0) \tag{1}$$

but

$$A_i(\sigma, \tau^0) \neq A_i(\sigma', \tau^0); \tag{2}$$

i.e., iteration i is the first iteration in which the actions of P_1 differ. (Note that some such τ^0, i must exist or else $\sigma = \sigma'$.) Without loss of generality, assume $A_i(\sigma, \tau^0)$ is to defect and $A_i(\sigma', \tau^0)$ is to cooperate.

Consider the following strategy τ of P_2: (1) act identically to τ^0 through iteration $i-1$; (2) in iteration i, defect; (3) in all subsequent iterations: if P_1 defected in iteration i, then cooperate; if P_1 cooperated in iteration i, defect. It is fairly immediate that $U_1(\sigma, \tau) > U_1(\sigma', \tau)$.

Next consider the following strategy τ': (1) act identically to τ^0 through iteration $i-1$; (2) in iteration i, cooperate; (3) in all subsequent iterations: if P_1 defected in iteration i, then defect; if P_1 cooperated in iteration i, cooperate. Exactly as when we argued earlier that our protocol was a Nash equilibrium, we have $U_1(\sigma, \tau') < U_1(\sigma', \tau')$.

The same argument extends to the case of general t, n. We simply replace τ^0 with a strategy profile of $n-1$ strategies such that Equations (1) and (2) above are still valid, and then define τ and τ' as above, but modifying the strategies of all other players. ■

Our protocol has no additional equilibrium which is preferred, by any player, to the prescribed equilibrium *assuming all communication occurs over the broadcast channel*. (This is because no covert channels exist in our protocol, assuming the dealer signs the distributed shares using a strong signature scheme.) On the other hand, if side communication between players (undetectable by other players) is possible then there is an equilibrium in which t players collude and reconstruct the

secret for themselves only. Note, however, that the Halpern-Teague protocol also has an undesirable equilibrium of this sort which is even worse: in their case, the 3 group leaders can pool the shares sent to them by all other parties and reconstruct the secret for themselves only. That is, their protocol is susceptible to coalitions of size 3, whereas ours is only susceptible to coalitions of size t.

3.3 Discussion

Our approach has a number of advantages as compared to the Halpern-Teague protocol:

- Most obvious, we circumvent their impossibility result for the case $n = 2$.
- Our protocol is much simpler than the Halpern-Teague protocol. This is true for all settings of t, n, but is especially true for the case of $n > 3$ (where the Halpern-Teague protocol requires players to coordinate and select group leaders) and/or the case $t = 2$ (whose solution is even more complicated; see [10]).
- Our protocol requires only a broadcast channel, in contrast to the Halpern-Teague protocol which relies on private channels in addition to broadcast.
- Each iteration in our protocol is more round-efficient than in the Halpern-Teague protocol. Furthermore, depending on the exact utility functions assumed, the expected number of iterations of our protocol is also lower.

4 Removing the Dealer

A drawback of both our protocol (as described in the previous section) as well as that of Halpern and Teague is that the dealer must be involved at the beginning of every iteration. It would be much nicer to have a solution that works exactly like standard secret sharing, where the dealer is involved only once at the beginning of the protocol.

We sketch here a conceptually simple (though inefficient) way to avoid continual involvement of the dealer while still ensuring that parties eventually reconstruct the secret with probability 1. Our idea applies both to our protocol and that of Halpern and Teague, but for simplicity we describe it in the context of our protocol only. The protocol proceeds as follows:

Setup: To share a secret s, the dealer prepares a valid t-out-of-n Shamir sharing $\{s_i\}$ of s. The dealer also generates a signature σ_i on each share s_i with respect to a publicly-known verification key PK (alternately, PK can simply be sent to each player). Finally, the dealer sends (s_i, σ_i) to player P_i.

The protocol: At the beginning of each iteration, the players proceed as follows:

1. The t participating parties run a secure computation protocol [13, 5, 4] to take the place of the dealer. The protocol should be secure against $t - 1$ malicious adversaries.[3] The protocol computes the following probabilistic functionality:
   - Each party inputs the values (s_i, σ_i) received from the dealer. The functionality checks that it receives t input values with each such value s_i properly signed with respect to PK, and aborts if this is not the case.

[3] Even if the protocol tolerates only *one* malicious party the resulting protocol will be a Nash equilibrium. However, we do not lose anything by requiring security against $t - 1$ players.

- The t input shares define a secret s. With probability β, the functionality generates a fresh t-out-of-n Shamir sharing $\{s'_i\}$ of s, and each player P_i receives output s'_i.
- With probability $1-\beta$, the functionality generates a fresh t-out-of-n Shamir sharing $\{s'_i\}$ of a bogus secret $\hat{s} \in \mathbb{F} \setminus S$, and each player P_i receives output s'_i.

2. If cheating is detected in the secure computation protocol above (i.e., the secure computation protocol is aborted), then parties terminate the overall protocol without ever reconstructing the secret.

3. Next, parties proceed as in the earlier protocol; specifically, each player P_i broadcasts the output s'_i they received from the secure computation protocol.[4] If this enables reconstruction of a secret $s \in S$, the protocol terminates and the true secret has been reconstructed. If some player refused to broadcast their output share, then parties terminate the protocol without reconstructing the secret. In any other case, players proceed to the next iteration.

Under the same conditions on β as before, following the above protocol is a Nash equilibrium surviving iterated domination of weakly dominated strategies.

5 General Secure Function Evaluation

The techniques outlined above generalize to the case of the secure computation of arbitrary functions. In this sense, they allow us to obtain a protocol for completely fair SFE under the assumption that (1) all players are rational; and (2) players' utility functions are such that they all prefer to learn the output. (In contrast, the work of [7] shows a protocol for completely fair SFE tolerating malicious players, but under a stronger assumption on the available communication. See Section 1.2.) In more detail, to compute the (possibly randomized) single-output function f the players will proceed as follows:

1. Let f' be the following (multi-output, randomized) function: on inputs $x_1, \ldots, x_n$, compute $y \leftarrow f(x_1, \ldots, x_n)$. Then generate a random t-out-of-n Shamir sharing $(s_1, \ldots, s_n)$ of the result y, and give output s_i to player P_i.

2. Players run a secure computation protocol for f', and obtain outputs $s_1, \ldots, s_n$. If this protocol is aborted, all players terminate the entire protocol and the output is never reconstructed.

3. Now, in a sequence of iterations as before, players compute a functionality that takes as input[5] $(s_1, \ldots, s_n)$ and, with probability β computes a random Shamir sharing $\{s'_i\}$ of the value y these shares define, and with probability $1-\beta$ computes a random Shamir sharing $\{s'_i\}$ of some default value not in the range of f. Each player P_i receives output s'_i. If this protocol is aborted, all players terminate the entire protocol and the output is never reconstructed.

4. Players simultaneously broadcast the s'_i and reconstruct the value s' these shares define. If some player did not broadcast a (valid) share, then all players terminate the protocol and do

[4] Actually, to prevent players from broadcasting a modified version of s'_i, it is necessary to change the secure computation protocol in some way; one easy way to do this (though others are possible) is to have the dealer initially distribute shares of his secret signing key in a t-out-of-n manner among the players. Then the secure computation protocol can also generate valid dealer signatures on the $\{s'_i\}$ values (the iteration number should also be signed to prevent replay of an earlier output value). We have omitted this for simplicity.

[5] As before, there is the issue of authenticating the shares $s_1, \ldots, s_n$ provided as input to this functionality. This can be handled in a similar manner as before.

not participate in any future iterations. If s' is in the range of f then $y = s'$ is the desired output and the protocol is done; in any other case, players proceed to the next iteration.

We remark that, as in standard formulations of secure multi-party computation, players who choose not to follow the protocol may change their "true" inputs to an arbitrary other value. (I.e., a player P_i with "true" input x_i may cause $f(x_1, \ldots, x_i', \ldots, x_n)$ to be evaluated for arbitrary x_i'.) For rational players, this may occur if a player would prefer to change his input value even if a completely incorruptible third party were to evaluate f based on inputs given to it by the players. For so-called *NCC functions* [9], however, it will be in rational players' best interests to enter their true inputs into the protocol.

6 Conclusions and Open Questions

We have provided a new approach to rational secret sharing that improves, in many respects, on an earlier solution of Halpern and Teague and which also improves upon the efficiency of the generic solutions of [7, 6]. We have also shown how our ideas extend to give a protocol for completely fair SFE in a more limited sense than that achieved by [7], but based on weaker assumptions on the communication between parties.

It would be nice to design a protocol that does not require knowledge of players' exact utility functions (recall that here such knowledge is used to compute an appropriate value of β), or that works with more general classes of utility functions than those considered here. Another question of interest is to investigate what happens if side communication between players is allowed; this is related to defending against deviations by *coalitions* of players.

Acknowledgments

We thank Silvio Micali for clarifications regarding the relationship between our work and [7, 8, 6], as well as for prompting us to think about using our techniques to obtain a more limited form of completely fair SFE.

References

[1] G.R. Blakley. Safeguarding Cryptographic Keys. *National Computer Conference*, vol. 48, pp. 313–317, AFIPS Press, 1979.

[2] R. Cleve. Limits on the Security of Coin Flips when Half the Processors are Faulty. *18th Annual ACM Symp. on Theory of Computing (STOC)*, 1986.

[3] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[4] O. Goldreich. *Foundations of Cryptography, vol. 2: Basic Applications*, Cambridge University Press, 2004.

[5] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. *19th Annual ACM Symp. on Theory of Computing (STOC)*, 1987.

[6] S. Izmalkov, S. Micali, and M. Lepinski. Rational Secure Function Evaluation and Ideal Mechanism Design. FOCS 2005.

[7] M. Lepinski, S. Micali, C, Peikert, and A. Shelat. Completely Fair SFE and Coalition-Safe Cheap Talk. *PODC* 2004.

[8] M. Lepinski, S. Micali, and A. Shelat. Collusion-Free Protocols. *37th Annual ACM Symp. on Theory of Computing (STOC)*, 2005.

[9] Y. Shoham and M. Tennenholtz. Non-Cooperative Computing: Boolean Functions with Correctness and Exclusivity. *Theoretical Computer Science* 343(1–2): 97–113 (2005).

[10] J. Halpern and V. Teague. Rational Secret Sharing and Multiparty Computation. *36th Annual ACM Symp. on Theory of Computing (STOC)*, 2004.

[11] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[12] A. Shamir. How to share a secret. *Comm. ACM*, 22(11): 612–613 (1979).

[13] A. C.-C. Yao. How to Generate and Exchange Secrets. *27th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, 1986.