Purely Rational Secret Sharing (Extended Abstract)

Silvio Micali[1] and abhi shelat[2]

[1] MIT CSAIL
silvio@csail.mit.edu
[2] U. Virginia
abhi@virginia.edu

Abstract. Rational secret sharing is a problem at the intersection of cryptography and game theory. In essence, a dealer wishes to engineer a communication game that, when rationally played, guarantees that each of the players learns the dealer's secret. Yet, all solutions proposed so far did not rely solely on the players' rationality, but also on their *beliefs*, and were also *quite inefficient.*

After providing a more complete definition of the problem, we exhibit a very efficient and purely rational solution to it with a verifiable trusted channel.

1 Introduction

In [LMPS04], Lepinski, Micali, Peikert and shelat put forward the notion and the first implementation of *Fair Secure Function Evaluation* . This is a communication protocol extending the traditional notion of secure function evaluation [GMW87]. In essence, a Fair SFE is an SFE in which either (1) all players learn the result of evaluating a given function on their secret inputs (but no other information about their inputs) or (2) none of them learns anything. The first outcome is reached when all players want it, and the second one when at least one of the players wants it. The difficulty lies in the fact that such objectives must be reached no matter what the function may be and no matter how many the player are, provided that at least one of the players is *honest*, that is sticking to his communication instructions in all cases.

In [HT04], Halpern and Teague put forward the notion of *rational secret sharing* (RSS), aiming at distilling separately, and in purely game theoretic terms, the last stage of a Fair SFE (where the players attempt to reconstruct the specified output from their shares of it). We believe this to be a very valuable contribution, but we also believe that the notion of an RSS can be improved.

In this extended abstract we shall solely deal with the two-player version of the notion, arguably the best way to highlight the novel and most poignant aspects of the problem.[1]

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-00457-5_36

[1] Our approach easily extends to n players, where the dealer wishes that n out of n of them learn the secret. The k-out-of-n definition of traditional secret sharing is very relevant for robustness, and protects against the potential loss of shares, but is quite distracting and orthogonal to the rationality problem at hand. Indeed, in a "k-out-of-n" rational secret sharing (assuming as usual that the fewer the players knowing the secret, the more value to them), k players will presumably prevent the others from learning their secret. Is this the natural wish of the dealer?

O. Reingold (Ed.): TCC 2009, LNCS 5444, pp. 54–71, 2009.

1.1 Rational Secret Sharing as a Special Form of Mechanism Design

The Intuitive Notion. At the verge of dyeing, a *dealer* possessing a secret string S wishes to ensure that two players will later on cooperate so as to *both* learn S. To this end, he provides each player i with his share of the secret, a string S_i. Each share is individually meaningless (i.e., its distribution is independent of S), while together the two shares reveal S. If the players were both honest, the dealer's goal could be trivially achieved. Unfortunately, honesty is not an available commodity: each player is assumed to be *rational* (i.e., always trying to maximize his own utility), and the utilities that the players attach to the possible ways of learning S are quite problematic. In particular, each player prefers most to be the only one learning S, prefers less learning S together with the other player, and even less not learning S at all. Accordingly, the dealer wishes to chose the shares such that,

> for a suitable communication channel, there exists a communication game that, when rationally played, yields the secret to both players.

It is thus worth to recall quickly the traditional notions of solving a game.

Game Solution Concepts. Given a game G, a solution concept essentially is a way of predicting how G will be played. From the cryptographic perspective of the authors, traditional solution concepts are only partially meaningful, as they are stated from the perspective of *individual* players, disregarding *collusion* altogether. Nonetheless their meaningfulness is intact for the problem at hand restricted to just two players. (This is by itself a good reason to focus on the two-player case.)

The strongest, traditional solution concept is that of solvability in *dominant strategies.* Here, each player has a strategy σ_i that is best of him, no matter what strategies the other players may use. In such a case predicting that each player i will play σ_i is indeed the strongest form of prediction of play. Note that, in choosing σ_i, each player i does not need to rely on the rationality on the other players, but just be rational himself! Unfortunately, not all games admit dominant-strategy solutions.

The "next best" solution concept is *dominant solvability,* which now very roughly explain. In a game G, a strategy a for player A is said to weakly dominate another strategy a' for A if (1) for all possible strategies b of player B, A's utility under a is greater than or equal to his utility under a'; and (2) for at least one strategy b' of B, A's utility under a is strictly greater than his utility under a'. This being the case, a rational A should remove strategy a' and all of his weakly dominated strategies from consideration. And a rational B should do the same on his side. Trusting that B has done so, A should then eliminate from his remaining strategies all those that now become weakly dominated (relative to the strategies left over to B). And so on, until neither player can eliminate any more strategies. At that point, if A is left with a single strategy a and player B with a single strategy b, then G is called *dominant solvable*, and (a, b) is a very strong prediction for the way in which G will be rationally played. Notice, however, that this second solution concept is weaker than the first one, since each player must rely non only on his rationality, but also on that of his opponent. Again, not all games admit such a strong solution concept.

The next solution concept we wish to roughly recall is that a Nash equilibrium. This is a pair of strategies (a, b), such that a is the best strategy for A if he believes that B will play b (and symmetrically for B). The good news is that each game admits such equilibria, but the bad news is this is a *very distant third* among these solution concepts. The meaningfulness of a Nash equilibrium in fact depends not only on the rationality of both players, but also on their *beliefs*. Typically a game has a plurality of equilibria, often having symmetric payoffs, making it very uncertain to predict which of them will be played. Furthermore, the game may easily not end up in equilibrium at all. If A believes that equilibrium (a, b) will be played, while B believes (a', b'), then the strategy profile ultimately played may be (a, b') which needs not to be an equilibrium at all!

Mechanism Design. Very roughly said the goal of mechanism design is to engineer a game so that, "when rationally played", a given property P is guaranteed to hold. The quality of such design therefore crucial depends on the solution concept adopted: it is exceptionally meaningful (recall that we are focusing on the two-player case!) when the game has dominant strategy solution, it is very meaningful when the game is "dominant solvable", and it has only very limited meaningfulness when the game is "Nash solvable." Such limited meaningfulness persists even if P is *guaranteed to hold at each of possible Nash equilibria of the game*. In a sense, if the game has k Nash equilibria, then —due mismatched beliefs— it roughly has k^2 (k^n if there are n players) possible ways not to end in any equilibrium.

Another quality measure in mechanism design is the amount of knowledge about the players (e.g., knowledge about their utilities) required to engineer the game. Indeed, since precise knowledge about the players may not be available or too expensive to gather, the lesser the knowledge required from the designer the better.

Rational Secret Sharing and Mechanism Design. We propose to view *rational secret sharing as a special mechanism-design problem*. That is, one should try to guarantee the property "all players learn the secret" by means of a pure communication game. In essence, the game should be such that (1) all player actions consist of exchanging messages over a special channel, (2) no trusted party is involved, and (3) no exogenous punishments, fines, etc. can be triggered by the final outcome: the players' utilities must solely depend on who has, or has not, learned the secret.

This point of view enables us to extend to RSS the same quality analysis applicable to mechanism design, providing a more meaningful comparison among various RSS protocols.

1.2 Prior Solutions

In their quoted paper, Halpern and Teague present a protocol for the 3-out-of-3 case (and then show how to modify it for the m-out-of-n case, where $3 \leq m < n$). Their protocol guarantees that all players learn the secret at a Nash equilibrium whose strategies survive the iterated elimination of weakly dominated strategies (IEOWDS for short). Rather than the *swap channel* of Lepinski, Micali, Peikert, and shelat, they rely on *simultaneous-broadcast channels*, and prove that no rational secret sharing protocol can be fixed-round with such channels. A main limitation of their protocol is that the trusted dealer continues to be an active participant. (In most settings, such a dealer could directly inform the players of what the secret is.)

Gordon and Katz [GK06] present a protocol for just two players that dismisses the need for the periodic involvement of the dealer. Their protocol too relies on simultaneous-broadcast channels, and guarantees that all players learn the secret at a Nash equilibrium whose strategies survive IEOWDS. Abraham, Dolev, Gonen, and Halpern [ADGH06] present a similar protocol, but focus on defining (and protecting against) coalitions of rational players.

Lysyanskaya and Triandopoulos [LT06], with the same channels and implementation type, consider a model with a mix of rational and malicious players.

Kol and Naor [KN08b] present a quite different protocol with simultaneous-broadcast channels, which guarantees that all players learn the secret at a "strict Nash equilibrium", a locally stronger version of a Nash equilibrium. (In essence, any player deviating from his own equilibrium strategy expects to receive a strictly smaller utility.)

A Separate Protocol. We wish to mention an interesting and recent protocol of Ong, Parkes, Rosen, and Vadhan [OPRV08]. Their protocol however works in a quite different model. On one side, it does not require any special channels (that is, it relies on ordinary broadcast channels rather then simultaneous-broadcast ones). On the other, it relies on the *honesty* of a few players. (As we focus solely on rational players, we shall not include this protocol in any future discussion or comparison.)

1.3 Weaknesses of Prior Solutions

Protocol Inefficiency and Excessive Designer Knowledge. The prior protocols share the following logical structure. The players interact in several rounds, using some special channels. The protocol has a special round R, unknown to the players because it is secretly selected by the dealer according to a given distribution. If no player "cheats" then all players learn the secret. A player can successfully cheat only if he correctly guesses R. If a player i erroneously guesses R, then no one learns the secret (which gives i utility u_i). But if i guesses R correctly (and acts appropriately), then he is the only one to learn the secret (which gives him utility U_i).

In essence, therefore, to hope that it is rational to stick to the protocol's prescribed strategies without cheating, letting p be the probability of successfully guessing R, p needs to be so small that $p \cdot u_i \leq (1-p)U_i$. This shows two separate weaknesses of these protocols. First, because properly engineering the game implies properly selecting p, the designer needs to know the u_i's and the U_i's quite accurately. (Thus, from a mechanism design perspective, this diminishes the quality of these approaches.) Second, because the expected number of rounds must be greater or equal to p, this implies that all prior protocols run in exponential time. In fact, independent of the distribution according to which R is selected, the expected number of rounds of the prior protocols must be exponential in k, assuming as it is natural that all players utilities are presented in binary, and that their length is k. This inefficiency alone calls for new protocols.

Limited Guarantee for the Desired Property. Prior solutions ensure that the property "all players learn the secret" holds at a given Nash equilibrium of the engineered communication game. Again, however, this assurance is far from guaranteeing our property for two separate reasons: *equilibrium selection* and *equilibrium absence*. Let us discuss the first reason first. Even if one were certain that the engineered game will end up in an

equilibrium, he could not be certain of which equilibrium would be actually selected. And since, in the engineered games of the previous works, the "all-know-the-secret" property was guaranteed only at one of the possible Nash equilibria, the equilibrium ultimately selected could very well be one in which not all players learn the dealer's secret. Let us now discuss the second reason. The meaningfulness of any Nash equilibrium is inextricably linked to the *assumption* that the players' beliefs are *"consistent",* which of course needs not be the case. Thus, *even if* all players learned the secret at each Nash equilibrium of the engineered games, there is no guarantee at all that the engineered game ends up in equilibrium. Again: assume that (a, b) and (a', b') are Nash equilibria, that A believes that B will play b, and that B believes that A will play a'. Then, A will rationally (based on her belief!) play strategy a, and B will play b'. And since (a, b') may not be an equilibrium, let alone an outcome in which all players learn the secret.

To be sure, the prior protocols were engineered so that all players learned the dealer's secret not just at a generic Nash equilibrium, but at one whose strategies survived IEOWDS. But as long as multiple Nash survive IEOWDS (which is the case in prior protocols), then equilibrium selection and equilibrium absence will continue to poison the landscape.

To be sure too, some of the prior RSS protocols guaranteed that all players learned the secret at an even stronger type of equilibrium, such as the strict Nash of [KN08b]. But these equilibria are in a sense only "locally stronger." That is, if the players believe that a strict Nash equilibrium (a, b) will be played out, they would have "even less incentives" of deviating from it. But ensuring that A does not deviate from a if she believes that B chooses strategy b is not too meaningful, unless one can also ensure that B actually chooses b. If the game is engineered so that the best we can say about it is that it has a strict Nash equilibrium (at which the desired property holds) alongside with other additional equilibria, then equilibrium selection and equilibrium absence will continue to stand in the way.

In sum, *all prior RSS protocols did not solely depend on the players' rationality, but also on their beliefs.* Thus they could not guarantee that all players, if rational, learned the secret.

1.4 Our Contributions

Our contributions can be summarized as follows.

- *Modeling.* We put forward a more complete modeling of the RSS problem.
  In particular: we highlight the inputs available to the designer of protocol; provide a more comprehensive set of utilities —including the possibility of learning the wrong secret—; highlight the necessity of modeling RSS as a potentially infinite communication game; provide a very general definition of a communication channel; highlight the necessity of worrying about other channels even in a communication game designed for a specific channel; provide the first rationalization of aborting in a communication game; and bring to the fore the necessity of including bargaining into the definition of RSS.

- *Purely Rational Implementation.* Our RSS protocol is an *implementation in surviving strategies*, as put forward by Chen and Micali [CM08]. In essence, such an implementation is "equilibrium-less." It guarantees that the desired property holds for any combination of strategies surviving IEOWDS. Implementation in surviving strategies thus implies that *the desired property is guaranteed based solely on the rationality of the players, and not on their beliefs.* In a sense, as long no player chooses a dumb strategy, the desired property is guaranteed to hold.

  Actually, our protocol satisfies a stronger notion of implementation: namely, the surviving strategy of each player is essentially *unique*.[2] That is, in our RSS protocol, after iteratively deleting all weakly dominated strategies, essentially a single strategy survives for each player, and playing these two strategies guarantees that both players learn the secret. That is, *our RSS protocol essentially is a dominant solvable game.*

  Note that IEOWDS often eliminates very few strategies (a fact that has been used to argue that Nash equilibria that survive IEOWDS is a solution concept not really better than an ordinary Nash). Thus it is even more remarkable that our protocol is such that, for any player, all but one strategy is "rationally credible."

  Note too that, in general, which strategy survives depends on the order in which weakly dominated strategies are eliminated. In our case, however, the (essentially) unique surviving strategy of a player is the same irrespective of any possible elimination order. In sum, our solution concept is indeed very strong.

- *Communication Channel and Security.* Our communication channel uses only ordinary envelopes (as a way of temporarily and perfectly hiding a secret value) and the dealer's public key.

  The security depends on the ability of envelopes to perfectly hide their content and unforgeable digital signatures.

- *Operational Efficiency.* Ours is the first polynomial-time RSS protocol, fully accounting for all inputs. In fact, each surviving strategy requires a total of $10k$ envelope operations, $4kL$ bit operations, plus the time of verifying two signatures relative to k-bit public keys. Here L is an upperbound to the length of the binary representation (of the absolute value) of any of the players' utilities, and k is a security parameter. The security parameter k controls the probability that something goes wrong. (The probability of something going wrong is guaranteed to be exponentially small in k.)

  The dealer is required to perform a total of $4kL$ bit operations, to generating matching public and secret keys of a digital signature with security parameter k, and to produce two signatures relative to the selected public key.

[2] The reason that we do not say unique is that, as we shall argue, a pure communication game G should be modeled as a possibly infinite sequence of the same sub-game g. Thus, a strategy of any player in G actually consists of a sequence of strategies, $\sigma_1, \sigma_2, \ldots$, where σ_j is the player's intended strategy for jth copy of g, if reached. By saying that each player has an essentially unique surviving strategy in G we mean that any of his surviving strategies is of the form $s, \sigma_2, \ldots$, where s is fixed; that is first sub-strategy is the same for any surviving strategy of the player. And when all players play their first such strategies, G terminates.

- *Round Efficiency.* A play of our surviving strategies involves only 6 rounds (1 for the players, and 5 for the channel).

2 Selected Modeling Issues

Dealer Secret, Player Outputs, Player Utilities, and Designer Knowledge. For concreteness, we model the secret as a uniformly selected string of n bits. (Our protocol of course works for all kinds of other distributions as well.)

We assume that, upon termination, each player outputs either an n-bit string (interpretable as the player's guess for the dealer's secret) or the special symbol "?" (interpretable as the player's having no information about the secret). The protocol terminates when a prespecified stage is publicly reached, or when either one of the players *aborts,* that is stops communicating and for ever takes no further action —after setting his own output.[3]

We define an outcome of an RSS protocol to consist of three possible outputs for each player: (1) the correct secret the dealer, (2) the symbol "?", and (3) an incorrect string. We assume that each player prefers his outputs in this order, and prefers the inverse order for the outputs of the other player. That is, for each player i, denoting by K_i (for "i knows the secret") his first output, by W_i (for "i wrongly learns the secret) his third output, and by u_i his utility function, we assume that the utilities of the first player over the possible 9 outcomes are as follows:

$$u_1(K_1, W_2) \geq u_1(K_1, ?) \geq u_1(K_1, K_2) \geq$$
$$u_1(?, W_2), \geq u_1(?, ?) = 0 \geq u_1(?, K_2) \geq$$
$$u_1(W_1, W_2), \geq u_1(W_1, ?) \geq u_1(W_1, K_2).$$

Player 2's utilities are symmetrically defined. (Setting the players' utilities to 0 when both of them have no information about the secret is somewhat arbitrary, but concretely useful to fix our thoughts.) All of the above inequalities can be strict. But for our analysis it suffices that $u_1(K_1, ?) > u_1(K_1, K_2) > u_1(?, ?) > u_1(W_1, W_2)$, and symmetrically for player 2. That is, each player prefers learning the secret alone to learning together with the other player, prefers the latter outcome to not learning the secret, and prefers the latter outcome to learning the wrong secret.[4] It is also useful to assume that a player's expected utility when randomly guessing the secret is negative. (Alternatively, we must ensure that the utility of random guessing the dealer's secret is less than that of learning the secret together with the other player. Else, a player would not have any incentive to participate in an RSS protocol.)

[3] That is, we explicitly assume that one players' aborting is detectable by the other player. (After all, stopping all communications should be "eventually detectable" in practical settings, and immediately detectable in synchronous ones.) Alternatively, each player may keep track of his current output at all times (rather than producing his output at termination). This way if a player aborts without the "courtesy" of informing the other player, the latter' output is properly set.

[4] Indeed, if the secret were the combination of a safe with money and a bomb inside, and the safe exploded when the wrong combination were entered, learning the wrong secret could have truly negative utility for a player!

This structure of the utility is assumed to be known to the designer. And so is an upperbound to the number of bits necessary to write down the largest of the 16 possibilities of the players. (In other words, it suffices for the designer to know the players' utilities within an exponential accuracy, rather than the linear accuracy of the prior works.)

Ensuring the Rationality of Abort. Our protocol, if a special point in which a player i has not yet learned the secret is reached, calls for him to abort. By so doing, of course, the player looses any hope of learning the secret. Thus, in order to guarantee that the suggested strategy survives IEOWDS, we need to ensure that, at that point of our protocol, the player no longer has any rational hope of learning the secret (whether alone or together with the other player). What should this mean? In particular, of course, it should mean that i's expected utility when continuing the current execution of the protocol is worse than that of aborting outputting "?". But it *should not* mean just that. The dealer who has provided the players with their shares is now dead, and can no longer control what the players do from his grave. RSS is a pure communication game, the players have all the information they need to continue any given execution of our protocol (if they so want) and no authority is there to stop them from (or fine them for) doing so. In addition, the players also have the ability of starting another execution from scratch. (For instance, they may use their same shares, but different coin tosses for their strategies, if probabilistic. Alternatively, if reusing the old shares is not "rationally advisable," they may first resort to a secure function evaluation to "compute new, equivalent and, independently selected shares from their old ones, and then execute our protocol again. The possible alternatives abound.) Better yet, perhaps, they also have the ability to start a totally different RSS protocol using the same communication channel. More generally yet (unless one were ready to make the outlandish assumption that no other channel exists), they have the ability to execute a totally different RSS protocol with a totally different channel! In sum,

> *To rationalize player i's aborting in an RSS protocol, we should* **prove** *that any chance of i's learning the secret has vanished.*

Realizing, formalizing, and delivering this property is a main contribution of our work.

Modeling Special Channels. All RSS protocols with rational players must use some special communication channel, such as a swap channel or a simultaneous-broadcast channel. Since we have just argued that a proper analysis of RSS should include the possibility of running a different protocol over a different channel, it becomes imperative to model any possible special channel of communication. We do so by letting special channels consist of "mildly trusted parties in abundant supply." Let us explain. If some party T could be totally and universally trusted, then many problems (including rational secret sharing) would be trivialized. For instance, the dealer might as well confide his secret to T and ask him to reveal it when all the designated players show up together. Thus "mild trust" became imperative. As for abundant supply we mean that there is not a unique mildly trusted party in the world. (If this were the case, one might ask T to interact only once with a given group of players for a given task, and simplify a lot of things too.) By contrast, to model the fact that a special communication channel (if it exists at all) is indeed a commodity purchasable at any store, we envisage that

there is a plurality of mildly trusted parties, not aware of —or not in contact with— each other.

Accordingly, following [ILM08], we model a mildly trusted party in abundant supply as a *verifiable trusted party* (VTP for short) with no memory. By verifiable we mean that every one can see the actions a VTP takes and verify that they are the prescribed ones. That is, a VTP is not trusted to keep, nor to correctly make any secret actions. A VTP knows nothing and acts publicly, so that he is trusted only to the extent that he will indeed publicly perform his prescribed public actions.

For example, a VTP can trivially implement a swap channel between two parties as follows. First each of A and B seals his message for the other into an opaque envelope and publicly gives it to the VTP. Then the VTP publicly hands A's envelope to B and B's envelope to A.

As for another example, a VTP can implement a simultaneous-broadcast channel as follows. First, A and B seal their respective messages for the other in two envelopes and publicly hand them to T, then T publicly opens both of them.

In sum, VTPs can be viewed as a formalization of a legal system. One may not want to trust his secrets to —say— a judge, but should at least trust a judge to carry out under public scrutiny a specified sequence of totally public actions. Since typically there are multiple judges to choose from, the analogy with the legal system makes it clear that the players can always walk to an new judge to execute their protocol one more time. The analogy also makes it clear that if one type of channel is available, then indeed other types are likely to be available too. Whether or not, as *functions*, the "swap channel" is reducible to the "simultaneous-broadcast-channel" (or viceversa), from the VTP perspective, both exist. (Indeed any judge can, with envelopes, implement both channels and a host of similar ones.) This highlights the point that when a player is asked to abort, then it really must be the case that no hope to resurrect the secret exists for him, no matter what other protocol and channel might be considered.

Adding Costs to the Model. Consider a cryptographic rational secret sharing protocol in which the dealer also announces an encryption E of the secret S. Then, a player, in addition to any other strategy, also has available a computational-attack one: namely, abort and try to decrypt E. A computational-attack strategy is also possible in our protocol, but in a more complex way. Indeed, successfully forging a given value enables a player to learn the secret alone, and force the other to learn a false secret. Thus we too need to argue that computational-attack strategies are not rational. One way to do so is to define a computationally bounded version of rational secret sharing. A preferable way is to *attach cost to computation* so as make it preferable for a player to play honestly our protocol rather than try to attack the signature scheme and then, if the attack is successful, getting an advantage in the protocol. Details will be provided in the final version. (In any case, as argued by Halpern and Pass [HP08] considering computational costs may be meaningful even for more traditional —i.e., non-cryptographic— game theoretic settings.)

We also associate a small additive cost of γ to each use of the channel. (E.g., every one has the right to access the legal system, but incurs a fixed cost in doing so.

We note that additive (or multiplicative) discounts of the players final utilities are quite standard in game theoretical models in which the players could go on interacting (possible even for ever), typically by executing a given sub-game.[5]

The Issue of Bargaining. Finally, let us bring to the reader attention a point totally neglected so far. Traditionally, to guarantee the dealer's wish that all players learn the secret (at least when everyone behaves rationally), the only restrictions envisaged for the utilities are *local to each player* (e.g., each player must prefer reconstructing the secret alone to reconstructing it together with the other player, etc.). That is, the utilities of an individual player must be "compatible with each other," but not with those of other players. We wish to point out, however, that it is necessary to consider *inter-player* restrictions on the players' utilities, or be ready live with the consequences relative to the dealer's wishes. Let us explain.

A dealer providing players with shares of his secret S automatically enables them to *bargain*. In a bargaining situation, one player may get a better deal than others *without any failure of rationality*. For instance, in an RSS context, Player 1 may simply insist that unless everyone plays a protocol in which he learns the secret *alone* 99 times out of 100, he is not going to cooperate. (In a sense, if to Bill Gates learning the secret together with you and me is worth \$1K, but learning it alone is worth \$1B, then he would be wasting time and opportunity costs in participating with you and me in a "fair" reconstruction of the secret. Therefore, he may successfully bargain for a higher probability of learning the secret.) Now, if the dealer indeed has come up with shares and channels enabling the players to rationally reconstruct the secret together using a given special communication channel, then we should also expect that —whether with the same or with a different channel— the players can use their same shares to skew the payoffs so as to suit their bargaining needs. Truly unbelievable assumptions must be made to prevent the shares to be used in this alternative manner (especially in light of the result of [ILM08], that essentially enables the players to do rationally almost anything, although not too conveniently). Thus, either one must make the additional assumption that the players utilities are such that their bargaining game has a unique solution (e.g., some form of symmetry), or the dealer must be ready to die in peace with the comfort that either all players (if rational) will learn the secret, or that he has put all of them on a *technically* equal bargaining position.

The reader is free to pick the assumption he prefers. But always guaranteeing that all players together learn the secret may not be possible. For the rest of this extended abstract let us assume that the utilities are such that there is a unique bargaining solution.

3 Our Enriched Solution

It is simpler to explain our protocol assuming first that also special, *dealer-sealed*, envelopes are available: anyone can verify that such an envelope has been sealed by the dealer, and thus that its content is what the dealer wanted it to be, because any attempt to break the dealer's seal is guaranteed to be detectable by anyone.

[5] For instance, if a given contract is executed after i days of negotiation it is worth less to the players than executing the same contract as $i - 1$ days of negotiation.

Notice that, if such special envelopes were available, then a trivial solution to the RSS problem exists. In essence, letting s be the secret, the Dealer creates two random strings s_A, s_B such that $s = s_A \oplus s_B$, and then provides player A (respectively B) with infinitely many pink (respectively, blue) dealer-sealed envelopes, each containing s_A (respectively s_B). Players A and B then interact as follows. First, each player, simultaneously with the other, gives the VTP one of his dealer-sealed envelopes. Then, if the VTP receives both a pink and a blue dealer-sealed envelope, he publicly opens both of them. Else (e.g., one of the envelopes is ordinary, or has a broken seal), he destroys all envelopes received. In either case, the players incur a positive cost for this interaction.

The above indeed is an RSS protocol working in dominant-strategies. The fact that s becomes public is not a problem: the dealer could just give both players the same string r and choose s_A and s_B such that their bit-by-bit exclusive-or is $s \oplus r$. The problem is that we see no way of keeping its analysis by simulating its dealer-sealed envelopes with ordinary ones and digital signatures. We thus now describe a more complex protocol for which we can "simulate" dealer-sealed envelops as follows. Rather than handing to a player infinitely many dealer-sealed envelopes with content c, the dealer gives him a single digital signature of c, which then the player can —copy and— put into an ordinary envelope and give to the VTP as many times as necessary. (In the final version we shall prove that this simulation keeps our analysis essentially intact.)

In order to guarantee implementation in surviving strategies, our protocol critically introduces an asymmetry in the way the players are treated.

3.1 Dealer's Instructions

On input an ℓ-bit secret s and a security parameter k', do:

1. Choose a random string $\sigma \in \{0,1\}^\ell$ and compute $s' \leftarrow s \oplus \sigma$.
2. Choose a value k such that for all i
   (a) $u_i(K_1, K_2) > \left(2^{-k/2}\right) u_i(K_1, ?) + \left(1 - 2^{-k/2}\right) u_i(?, ?)$
3. For $i = 1, 2, \ldots, k$, repeat the following
   (a) Randomly select a four-tuple (a_0, a_1, b_0, b_1) such that a_0, b_0 are a random $\oplus$-sharing of the secret s' and a_1, b_1 are random and independent values of the same length as s.
   (b) Pick two random bits $e_1, e_2 \leftarrow \{0,1\}$.
   (c) Player 1's share is (a_{e_1}, a_{1-e_1}) and Player 2's share is (b_{e_2}, b_{1-e_2}).
   (d) Player 1's check value is $C_{1,i} = (e_2, b_1)$ and player 2's check value is $C_{2,i} = (e_1, a_1)$.
   (e) Place value a_j into envelope $E_{1,i,j}$ and place value b_j into envelope $E_{2,i,j}$ for $j \in \{0,1\}$.
4. Let C be the $k(\ell+1)$-bit number corresponding to the check values $C_{2,1}, \ldots, C_{2,k}$. Choose random values $\alpha, \beta \in \mathbb{Z}_k$ and compute the message authentication code $\gamma = \alpha \cdot C + \beta$.
5. Place into an envelope $E_{1,0}$ the values $(C_{1,1}, \ldots, C_{1,k}, \alpha, \beta)$ and into an envelope $E_{2,0}$ the values $(C_{2,1}, \ldots, C_{2,k}, \gamma)$. Seal the envelope $E_{1,0}$.
6. Place into an envelope $E_{p,\sigma}$ the value σ for $p \in \{0,1\}$.
7. Send the player 1 the envelopes $E_{1,0}, E_{1,\sigma}$ and $E_{1,i,j}$ for $i \in [1,k]$ and $j \in \{0,1\}$. Send to player 2 the envelopes $E_{1,0}, E_{1,\sigma}$ and $E_{1,i,j}$ for $i \in [1,k]$ and $j \in \{0,1\}$.

3.2 Reconstruction Instructions

Recall that a player's strategy consists of a Turing machine that on input a history h outputs either a special symbol $\perp$ to indicate abort, an output string s, or a sequence of $2k+1$ strings to place into envelopes that are submitted to the VTP. We use the symbol ε to denote the initial history consisting of only the envelopes received from the dealer.

Player p instructions $T(h)$:

1. If $h = \varepsilon$, then submit envelopes $E_{p,0}$ and $E_{p,i,j}$ for $i \in [1,k]$ to the VTP. If the VTP destroy the envelopes, output $\perp$ and stop. Else, after the VTP completes all of its steps, reconstruct n candidates of s by xor'ing the non-check values that have been opened. Let s' be the majority candidate. If no majority exists, then output $\perp$. Otherwise, privately open envelope E_σ and output $s' \oplus \sigma$.
2. For all other histories, output $\perp$ (i.e. do not invoke the VTP).

VTP Instructions:

1. Publicly verify envelope $E_{1,0}$. If the envelope's seal does not verify, then destroy all envelopes. Otherwise, publicly open the envelope to reveal the values $(C_{1,1}, \ldots, C_{1,k})$ and α, β.
2. Publicly open envelope $E_{2,0}$ to reveal values $C = (C_{2,1}, \ldots, C_{2,k})$ and γ. If $\gamma \neq \alpha \cdot C + \beta$, then destroy all envelopes.
3. Open the check envelopes (left or right) of player two indicated by $C_{1,i}, \ldots, C_{1,k}$. If there exists an opened envelope $E_{2,i,j}$ that does not match its stated value in $C_{1,i}$, the check fails: destroy all envelopes.
4. Open the check envelopes (left or right) of player one indicated by $C_{2,i}, \ldots, C_{2,k}$. If there exists an opened envelope $E_{1,i,j}$ that does not match its stated value in $C_{2,i}$, the check fails: destroy all envelopes.
5. If all k checks succeed, open the remaining $2k$ envelopes (corresponding to shares of the secret s').

3.3 Analysis

Theorem 1. *The strategy profile (T, T) for players 1 and 2 constitute a profile that uniquely survives the iterated deletion of weakly dominated strategies in the given VTP model.*

The main idea of the proof. Unless the first envelope submitted by the first player is sealed correctly, the VTP destroys envelopes. Once the one-and-only sealed envelope $E_{1,0}$ is opened, the second player knows which of her share values are check values, and which are values that are used in the sharing of s'. If the VTP succeeds in the same use that $E_{1,0}$ is opened, then both players learn the secret. If it does not, then some check envelope has failed and therefore no share value has yet been opened. In subsequent uses of the VTP, the second player can then modify all of her share values by XORing a random string r to them. This action is undetectable by the first player. Moreover, this action is the weakly dominant response for player 2 since player 2 prefers to learn the secret alone. Therefore, the first player has no hope to recover the secret (since any future opened share values will be independent of the real secret s'. Thus, the first player will abort in every subsequent use of the VTP. As a result, it is best for the second

player to submit the envelopes received from the dealer on the first use (since either her envelopes are never opened, or they are opened in the first and only rational opportunity there will be to recover the secret). In this case, the first player should follow T since each use of the VTP incurs a small cost. Then finally, the second player should also play T.

Definition 1. *A* revealing history h is a history in which the envelope $E_{1,0}$ has been opened and verified in some use of the VTP, but in every use of the VTP, all envelopes have been destroyed.

Let X_1 be the set of all player-one strategies, and X_2 be the set of all player-two strategies. Notice that for all $\sigma \in X_1$, $u_2(\sigma, T) \geq u_2(?, ?)$ and for all $\tau \in X_2$, $u_1(T, \tau) \geq u_1(?, ?)$. Therefore in the first step of removal, all guessing strategies that have expected utility less than $u_i(?, ?)$ can be removed.

For any player-two strategy τ, define $\Gamma(\tau)$ as the following strategy:

1. For the first use of the VTP, follow $\tau(\varepsilon)$. If the first use of the VTP results in all envelopes being opened, output the same as strategy τ.
2. If the first use of the VTP does not result in all envelopes being opened, for the subsequent uses of the VTP, follow strategy τ with the following exception: for any revealing history h, compute which of player 2's envelopes are non-check envelopes, choose a random value r and XOR r to each of these non-check values. Use these new non-check envelope values in place of the original non-check values received from the dealer to compute $\tau(h)$ for all subsequent histories h. If in this use or any subsequent use of the VTP, all envelopes are opened, compute the output O as per τ using the original non-check envelope values.

Claim. The player-two strategy $\Gamma(\tau)$ weakly dominates τ whenever $\tau \neq \Gamma(\tau)$.

For any player-one strategy σ, the player-two strategies τ and $\Gamma(\tau)$ are the same for the first use of the VTP, and thus result in similar utilities in any execution that succeeds.

For any revealing history, $\Gamma(\tau)$ never does worse than τ since $\Gamma(\tau)$ is both perfectly indistinguishable from τ to player one, and the share values produced by $\Gamma(\tau)$ do not have any information about the secret s'. Since $\Gamma(\tau) \neq \tau$, then there is some σ and some execution for which $\Gamma(\tau)$ will be strictly better than τ.
Set $X_2^1 = \{\Gamma(\tau)\}_{\tau \in X_2}$. For any player-one strategy σ, let $\Pi(\sigma)$ be the strategy that does the following: If the input history h is not revealing, then follow $\sigma(h)$. If input history h is revealing, then (a) never use the VTP in any subsequent round and (b) if $\sigma(h)$ outputs a string s, then output s and otherwise output $\perp$.

Claim. The player-one strategy $\Pi(\sigma)$ weakly dominates σ whenever (1) $\sigma(\varepsilon)$ submits the sealed envelope $E_{1,0}$, and (2) there exists $\tau' \in X_2^1$ such that (σ, τ') produces a revealing history h with positive probability and $\sigma(h)$ does not instruct to abort.

Consider any profile (σ, τ) were $\tau \in X_2^1$. The strategies σ and $\Pi(\sigma)$ are equivalent on the first use of the VTP and therefore result in the same history h. If h is successful, then both σ and $\Pi(\sigma)$ result in reconstructing the secret. Similarly, if h is not successful and also not revealing, then the two strategies are equivalent. If h is revealing, but $\sigma(h)$

produces an output, then both are equivalent. Finally, if h is a revealing history and $\sigma(h)$ uses the VTP again, then $\Pi(\sigma)$ is strictly better. This follows because τ survives the first step of removal, and therefore τ produces envelopes for the second (and future) uses of the VTP that are independent of the secret s'. This upper-bounds player 1's utility $u_1(\sigma, \tau)$ by $-\epsilon + u_1(?, \cdot)$. However, $u_1(\Pi(\sigma), \tau') = u_1(?, \cdot)$ which is strictly greater. (Similar analysis for the case when σ outputs s instead of $\perp$.)

The second condition of the claim ensures this situation occurs for some τ, and therefore therefore $\Pi(\sigma)$ weakly dominates σ.
Set X_1^1 to be the set of player-one stratgies in which after the sealed envelope is submitted, the VTP is never used again. Let $\Theta^i(\tau)$ be the player-two strategy that plays $T(\varepsilon)$ in the first i uses of the VTP, and follows τ for all subsequent uses.

Claim. If $\tau \neq \Theta^1(\tau)$, then $\Theta^1(\tau)$ weakly dominates τ.

Consider any player-one strategy $\sigma \in X_1^1$.

For those executions of σ in which player 1 submits an unsealed envelope in the first use of the VTP, all envelopes are immediately destroyed and therefore it holds that $u_2(\sigma, \Theta(\tau)) = u_2(\sigma, \tau)$ since both strategies are equivalent for all second and subsequent uses of the VTP.

We now consider those executions of σ in which the sealed $E_{1,0}$ is submitted. (This can only happen once.) Let $p_{\sigma,\tau}$ be the probability that under profile (σ, τ), the first use of the VTP results in destroyed envelopes. Observe that $p_{\sigma,\tau} \geq p_{\sigma,\Theta(\tau)}$ for all σ. Since $\sigma \in X_1^1$, the VTP is never used again by σ, and therefore $u_2(\sigma, \tau) = p_{\sigma,\tau} u_2(\cdot, K_2)$ which is less than or equal to $p_{\sigma,\Theta(\tau)} u_2(\cdot, K_2) = u_2(\sigma, \Theta(\tau))$. The condition that $\Theta(\tau) \neq \tau$ implies that the inequality is strict for some player one strategy σ which establishes the claim. Induction can be used to show that the claim holds for all i.
Set $X_2^2 = \{\Theta(\tau)\}_{\tau \in X_1^1}$.

Claim. The player-one strategy T weakly dominates every surviving strategy σ.

Observe that $u_1(T, \tau) = u_1(K_1, K_2)$ for any $\tau \in X_2^2$. Any other player one strategy has a positive probability of causing the VTP to destroy all envelopes, and therefore incurring a cost of $-\epsilon$.

A similar argument with Π can be applied to every player-two strategy. Thus, in any use of the VTP that reveals the dealer-received envelope E_2, the player-two strategy no longer uses the VTP. This implies that the player-two strategy T weakly dominates every surviving strategy.

Acknowledgements

Many thanks to Sergei Izmalkov for his characteristically generous and insightful help.

References

[ADGH06] Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. In: PODC 2006 (2006)

[BP98] Ben-Porath, E.: Correlation without mediation: Expanding the set of equilibria outcomes by "cheap" pre-play procedures. J. of Economic Theory 80, 108–122 (1998)
[CM08] Chen, J., Micali, S.: Resilient Mechanisms For Truly Combinatorial Auctions. MIT-CSAIL-TR-2008-067 (November 2008)
[GK06] Gordon, S.D., Katz, J.: Rational Secret Sharing, Revisited. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 229–241. Springer, Heidelberg (2006)
[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: STOC 1987 (1987)
[HT04] Halpern, J., Teague, V.: Rational secret sharing and multiparty computation. In: STOC 2004 (2004)
[HP08] Halpern, J., Pass, R.: Game Theory with Costly Computation (manuscript, 2008)
[IML05] Izmalkov, S., Micali, S., Lepinski, M.: Rational Secure Computation and Ideal Mechanism Design. In: FOCS 2005 (2005)
[ILM08] Izmalkov, S., Lepinski, M., Micali, S.: Verifiably secure devices. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 273–301. Springer, Heidelberg (2008)
[KN08a] Kol, G., Naor, M.: Cryptography and Game Theory: Designing Protocols for Exchanging Information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
[KN08b] Kol, G., Naor, M.: Games for Exchanging Information. In: STOC 2008 (2008)
[LMPS04] Lepinski, M., Micali, S., Peikert, C., Shelat, A.: Completely Fair SFE and Coalition-Safe Cheap Talk. In: PODC 2004 (2004)
[LT06] Lysyanskaya, A., Triandopoulos, N.: Rationality and adversarial behavior in multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
[OPRV08] Ong, S.J., Parkes, D., Rosen, A., Vadhan, S.: Fairness with an Honest Minority and a Rational Majority. On Eprint, 2008/097 (2008)
[OR] Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)

A The Ballot-Box Model

Ballot-box mechanisms are extensive-form, imperfect-information mechanisms with Nature. Accordingly, to specify them we must specify who acts when, the actions and the information available to the players, when the play terminates, and how the outcome is determined upon termination.

A ballot-box mechanism ultimately is a mathematical abstraction, but possesses a quite natural physical interpretation. The physical setting is that of a group of players, seated around a table, acting on a set of *ballots*. Within this physical setting, one has considerable latitude in choosing reasonable actions available to the players. In this paper, we make a specific choice, sufficient for our present goals.

A.1 Intuition

Ballots. Externally, all ballots of the same kind are identical. (Unlike [ILM08], we do not need super-envelopes here.) An envelope may contain a symbol from a finite alphabet. An envelope perfectly hides and guarantees the integrity of the symbol it contains until it is opened. Initially, all ballots are empty and in sufficient supply.

Ballot-Box Operations. We only need 3 classes of ballot-box operations. Each operation except for the first type is referred to as a *public action*, because it is performed

in plain view, so that all players know exactly which action has been performed. These classes are: (1) writing a symbol on a piece of paper and sealing it into a new, empty envelope; (2) publicly opening an envelope to reveal its content to all players; (3) publicly destroying a ballot; and (4) do nothing.

Public Information. Conceptually, the players observe which actions have been performed on which ballots. Formally, (1) we associate to each ballot a unique identifier, a positive integer that is common information to all players (these identifiers correspond to the order in which the ballots are placed on the table for the first time or returned to the table —e.g., after being ballot-boxed); and (2) we have each action generate, when executed, a public string of the form "A, j, k, l, ..."; where A is a string identifying the action and $j, k, l, ...$ are the identifiers of the ballots involved. The *public record* is the concatenation of the public strings generated by all actions executed thus far.

A.2 Formalization

Basic Notation. We denote by Σ the alphabet consisting of English letters, arabic numerals, and punctuation marks; by Σ^* the set of all finite strings over Σ; by $\mathbb{S}_k$ the group of permutations of k elements; by $x := y$ the operation that assigns value y to variable x; by $p := rand(\mathbb{S}_k)$ the operation that assigns to variable p a randomly selected permutation in $\mathbb{S}_k$; and by $\emptyset$ the empty set.

If S is a set, by S^0 we denote the empty set, and by S^k the Cartesian product of S with itself k times. If x is a sequence, by either x^i or x_i we denote x's ith element,[6] and by $\{x\}$ the set $\{z : x^i = z \text{ for some } i\}$. If x and y are sequences, respectively of length j and k, by $x \circ y$ we denote their concatenation (i.e., the sequence of $j + k$ elements whose ith element is x^i if $i \leq j$, and y^{i-j} otherwise). If x and y are strings (i.e., sequences with elements in Σ), we denote their concatenation by xy.

If A is a probabilistic algorithm, the distribution over A's outputs on input x is denoted by $A(x)$. A probabilistic function $f : X \to Y$ is *finite* if X and Y are both finite sets and, for every $x \in X$ and $y \in Y$, the probability that $f(x) = y$ has a finite binary representation.

Ballots and Actions. An *envelope* is a triple $(j, c, 0)$, where j is a positive integer, and c a symbol of Σ. A *ballot* is an envelope. If (j, c, L) is a ballot, we refer to j as its *identifier*, to c as its *content*, and to L as its *level*.

A set of ballots B is *well-defined* if distinct ballots have distinct identifiers. If B is a well-defined set of ballots, then I_B denotes the set of identifiers of B's ballots. For $j \in I_B$, B_j (or the expression *ballot j*) denotes the unique ballot of B whose identifier is j. For $J \subset I_B$, B_J denotes the set of ballots of B whose identifiers belong to J.

Relative to a well-defined set of ballots B: if j is an envelope in B, then $cont_B(j)$ denotes the content of j; if $x = j^1, \ldots, j^k$ is a sequence of envelope identifiers in I_B, then $cont_B(x)$ denotes the concatenation of the contents of these envelopes, that is, the string $cont_B(j^1) \cdots cont_B(j^k)$.

[6] For any given sequence, we shall solely use superscripts, or solely subscripts, to denote all of its elements.

A *global memory* consists of a pair (B, R), where

- B is a well defined set of ballots; and
- R is a sequence of strings in Σ^*, $R = R^1, R^2, \ldots$.

We refer to B as the *ballot set;* to R as the *public record;* and to each element of R as a *record*. The *empty global memory* is the global memory for which the ballot set and the public record are empty. We denote the set of all possible global memories by GM.

Ballot-box actions are functions from GM to GM. The subset of ballot-box actions available at a given global memory gm is denoted by $\mathcal{A}_{gm}$. The actions in $\mathcal{A}_{gm}$ are described below, grouped in 8 classes. For each $a \in \mathcal{A}_{gm}$ we provide a formal identifier; an informal reference (to facilitate the high-level description of our constructions); and a functional specification. If $gm = (B, R)$, we actually specify $a(gm)$ as a program acting on variables B and R. For convenience, we include in R the auxiliary variable $\mathtt{ub}$, the *identifier upper-bound*: a value equal to 0 for an empty global memory, and always greater than or equal to any identifier in I_B.

1. $(\textsc{NewEn}, c)$ —where $c \in \Sigma$.
   "Make a new envelope with content c."
   $\mathtt{ub} := \mathtt{ub} + 1$; $B := B \cup \{(\mathtt{ub}, c, 0)\}$; and $R := R \circ (\textsc{NewEn}, c, \mathtt{ub})$.
2. $(\textsc{OpenEn}, j)$ —where j is an envelope identifier in I_B.
   "Publicly open envelope j to reveal content $cont_B(j)$."
   $B := B \setminus \{B_j\}$ and $R := R \circ (\textsc{OpenEn}, j, cont_B(j), \mathtt{ub})$.
3. $(\textsc{Destroy}, j)$ —where j is a ballot identifier in I_B.
   "Destroy ballot j"
   $B := B \setminus \{B_j\}$ and $R := R \circ (\textsc{Destroy}, j, \mathtt{ub})$.
4. $(\textsc{DoNothing})$.
   "Do nothing"
   $B := B$ and $R := R \circ (\textsc{DoNothing}, \mathtt{ub})$.

Remarks

- All ballot-box actions are deterministic functions.
- The variable $\mathtt{ub}$ never decreases and coincides with the maximum of all identifiers "ever in existence." Notice that we never re-use the identifier of a ballot that has left, temporarily or for ever, the table. This ensures that different ballots get different identifiers.

Definition 2. *A global memory gm is* feasible *if there exists a sequence of global memories $gm^0, gm^1, \ldots, gm^k$, such that gm^0 is the empty global memory; $gm^k = gm$; and, for all $i \in [1, k]$, $gm^i = a^i(gm^{i-1})$ for some $a^i \in \mathcal{A}_{gm^{i-1}}$.*

If (B, R) is a feasible memory, we refer to R as a feasible public record.

Notice that if $gm = (B, R)$ is feasible, then $\mathcal{A}_{gm}$ is easily computable from R alone. Indeed, what ballots are in play, which ballots are envelopes and which are super-envelopes, *et cetera,* are all deducible from R. Therefore, different feasible global memories that have the same public record also have the same set of available actions. This motivates the following definition.

Definition 3. *If R is a feasible public record, by $\mathcal{A}_R$ we denote the set of available actions for any feasible global memory with public record R.*

B The Notion of a Public Ballot-Box Mediator (VTP in Our Language)

Definition 4. *Let $\mathcal{P}$ be a sequence of K functions. We say that $\mathcal{P}$ is a* public ballot-box mediator (of length K) *if, for all $k \in [1, K]$ and public records R, $P^k(R)$ is a public ballot-box action in $\mathcal{A}_R$.*

An execution of $\mathcal{P}$ *on an initial feasible global memory (B^0, R^0) is a sequence of global memories*
$(B^0, R^0), \ldots, (B^K, R^K)$ *such that $(B^k, R^k) = a^k(B^{k-1}, R^{k-1})$ for all $k \in [1, K]$, where $a^k = P^k(R^{k-1})$.*[7]

If e is an execution of $\mathcal{P}$, by $B^k(e)$ and $R^k(e)$ we denote, respectively, the ballot set, the public record, and the private history profile of e at round k. By $R^k_{\mathcal{P}}(e)$ we denote the last k records of $R^k(e)$ (i.e., "the records appended to R^0 by executing $\mathcal{P}$").

Remarks

- Note that the above definition captures our intuitive desideratum that no special trust is bestowed on a public mediator. Because he performs a sequence of public ballot-box actions, any one can verify that
  - (i) he performs the right sequence of actions;
  - (ii) he does not choose these actions; and
  - (iii) he does not learn any information that is not publicly available.
- Note too that if $\mathcal{P} = P^1, \ldots, P^K$ and $\mathcal{Q} = Q^1, \ldots, Q^L$ are public mediators, then their concatenation, that is, $P^1, \ldots, P^K, Q^1, \ldots, Q^L$ is a public mediator too.

[7] Note that the executions of $\mathcal{P}$ are, in general, random since $P^k(R)$ may return an action of Nature.