Rational Secret Sharing with Competition

Tiantian Gong[1⋆] and Zeyu Liu[2]

Yale University, New Haven, USA,
{tiantian.gong, zeyu.liu}@yale.edu

Abstract. The rational secret sharing problem (RSS) considers incentivizing rational parties to share their received information to reconstruct a correctly shared secret. Halpern and Teague (STOC'04) demonstrate that solving the RSS problem deterministically with explicitly bounded runtime is impossible, if parties prefer learning the secret than not learning, and they prefer fewer other parties to learn.
To overcome this impossibility result, we propose RSS with *competition*. We consider a slightly different yet sensible preference profile: Each party prefers to learn the secret *early* and prefers fewer parties learning *before* them. This preference profile changes the information-hiding dynamics among parties in prior works: First, those who have learned the secret are indifferent towards or even prefer informing others later; second, the competition to learn the secret earlier among different access groups in the access structure facilitates information sharing inside an access group. As a result, we are able to construct the *first* deterministic RSS algorithm that terminates in *at most* two rounds. Additionally, our construction does not employ any cryptographic machinery (being fully game-theoretic and using the underlying secret-sharing scheme as a black-box) nor requires the knowledge of the parties' exact utility function. Furthermore, we consider general access structures.

1 Introduction

Secret sharing is one of the fundamental building blocks of modern cryptography. An m-out-of-n secret sharing scheme shares one secret among n parties and then m people together can reconstruct this secret. In this access structure, any m parties form an *access group*. However, this threshold secret sharing notion naturally assumes at least m parties are honest (i.e., follow the protocol exactly) to ensure a successful reconstruction. Such an assumption can be strong in some applications. Therefore, Halpern and Teague [12] proposed the rational secret sharing (RSS) problem.

In particular, RSS only assumes that the participants are rational, and aims to achieve *fair reconstruction*. Informally, a fair reconstruction in an RSS protocol run by rational (and possibly also malicious parties) allows rational parties to all learn the secret. Here, a rational party follows some preference profile when

⋆ This work is mostly done while the author was a Ph.D. candidate at Purdue University.

taking an action. As an example, the preference profile in [12] is that rational parties prefer learning the secret over not learning and prefer that fewer others also learn the secret. Under this profile, it was shown in [12] that it is impossible to build a fair reconstruction that is *deterministic* and runs for a *known bounded number of rounds*. Intuitively, this is because in the last round, not sharing dominates sharing when $(m-1)$ other parties share. Then via backward induction, the Nash equilibrium that survives iterated deletion of dominated strategies from the last to the first round is each rational party sharing no information.

To circumvent such an impossibility result, past works have taken a few different routes: (1) adding randomness so that reconstruction terminates probabilistically, and parties can penalize those not sending shares in the previous round by aborting [12,17,1], (2) considering different preference profiles, e.g., only preferring learning [1], (3) considering infinite or unknown rounds of reconstruction, where backward induction no longer applies, and defecting can always be penalized, or (4) discussing repeated reconstructions for many secrets where parties can penalize cheating parties in later runs reconstructing a different secret [23]. Among these approaches, (1), (3), and (4) share the same intrinsic rationale: parties who do not honestly help reconstruction get penalized, which promotes cooperation. Following this idea, one may also consider RSS to be part of a larger protocol where penalties can be applied outside RSS executions. However, one issue with this method together with approaches (3) and (4) is that they are not as generic as (1) and (2). Thus, we avoid such non-generic settings.

For approach (1), previous works have explored randomized RSS protocols based on Shamir's secret sharing scheme [27]. The protocol typically proceeds in rounds. In each round, each party decides whether to send their correct share to others or not. Each round is an actual reconstruction round for the secret with some probability $\alpha \in (0, 1)$: If a round is not an actual reconstruction round, the secret is not successfully reconstructed, and parties start a new round. Parties abort if others deviate from sharing in the previous round. Otherwise, the protocol terminates when it hits an actual reconstruction round. Here, α is parameterized with parties' utility functions in a way such that rational parties are incentivized to share.

To design such a randomized RSS algorithm, past works have used some heavy tools, including simultaneous broadcast (SBC) [5] (or time-delayed encryption (TDE) [20], homomorphic time-lock puzzles (HTLP) [22], verifiable random function (VRF) [24]), along with a trusted mediator or a secure multiparty computation (MPC) protocol [10], and digital signatures [12,11,1,20,17,18,25,8,19,3] [15,16]. With these primitives and given a proper α, sending shares to others is made the equilibrium strategy (in the respective equilibrium in each past work). We give a more detailed description of these RSS protocols in Section 2 and a summary in Table 1. However, there are several major issues with these solutions: (1) they use heavy machinery and thus lack practicality; (2) α may be small, meaning that the protocol can potentially last many rounds (e.g., hundreds with $\alpha < 0.01$); (3) α is set according to the utility function of each party

and thus the function must be known explicitly. These issues are also detailed in Section 2.

An alternative preference profile. To avoid the above drawbacks, we choose the route of devising a different yet still sensible preference profile. In practice, information can be *time-sensitive*: People may not only prefer learning the information but also prefer learning it earlier than others, since this may give them advantages. For example, in stock markets, people who learn information earlier can front-run others and make a profit. Moreover, for people who have already learned and capitalized on the secret, they may still prefer others learning it than not. For instance, others buying the same stock after oneself increases the bid prices. On the other hand, people who have not learned still prefer learning than not learning. In the same example, not learning incurs opportunity costs.

With this intuition in mind, we consider a preference profile accounting for *time and competition*: Rational parties prefer learning the secret in earlier rounds and prefer fewer parties learning the secret before themselves. However, they still prefer learning it than not.

This profile especially applies to RSS in the following applications. (1) It applies where the secrets generate time-sensitive returns and can be capitalized fast, e.g., the reconstructed secret indicates investment opportunities. In this case, learning the secret earlier yields higher profits and one is indifferent towards informing others after acting on the information. (2) It also applies where parties benefit from others taking a certain predictable action after learning the secret, e.g., buying the same stocks as oneself. In this case, one prefers informing others after learning. (3) It also applies to applications where accomplishing computation tasks faster facilitates progress, e.g., distributed computing protocols [9] where a supermajority of parties reconstructing a secret– such as the randomness– early can form a committee to make progress.

1.1 Contributions

New preference profile. We first formally define the new preference profile that provides a new direction to circumvent the impossibility. Intuitively, under the updated preference profile, rational parties are naturally encouraged to share information inside their access groups to learn the secret quickly and before other access groups.[1] Indeed, we show that sharing information in the first or second round is the subgame perfect equilibrium (in the scheme below), which is *stronger* than the Nash equilibrium (NE) used in prior works.

Deterministic RSS construction. With this new preference profile, we provide a *deterministic RSS* scheme (with general access structure, see Section 3.2 for details) that terminates in two rounds without requiring simultaneous communication. In particular, it has the following advantages:

- No heavy cryptographic machinery: Major previous works [12,11,1,20,17,18] [25,8,19,3,15,16] focus on probabilistic termination of the protocol and avoid

[1] Except for the singleton access structure (i.e., all parties need to participate to recover the secret).

parties' gaining information advantages in the reconstruction game by either enforcing simultaneous movements with SBC or other heavy tools including timed primitives and VRF.

In contrast, we lift all these requirements: we do not even request broadcast channels and allow parties to move sequentially. [2] We also do not need a trusted mediator or MPC functioning as a mediator during reconstruction.

- Knowledge about the utility function is not required. The previous randomized protocols require the knowledge of participants' exact utility functions to parameterize α, the protocol termination parameter, properly. Such knowledge is *not* needed in deterministic reconstruction.
- Arbitrary side information does not affect the protocol. As pointed out by Asharov and Lindell [3], access to auxiliary information about the secret motivates deviation from the equilibrium of information sharing in randomized RSS protocols that do not rely on SBC. Intuitively, the informed party can potentially reconstruct and recognize the secret without sharing and entering the next round. Lysyanskaya and Segal [19] circumvent this by utilizing TDE to hide the reconstructed output for a sufficiently long time. Our updated preference profile allows RSS to accommodate side information because each round is needed for actual reconstruction (unlike prior works where each round is only for actual reconstruction with some probability), and auxiliary information does not give informed parties an advantage.
- Small complexity overhead. The reconstruction of our scheme terminates in constant rounds. It also has little communication and computation overhead, and the share size is *unchanged* from generic secret sharing schemes. Furthermore, when the randomness of the schemes relies on secret sharing schemes (e.g., distributed randomness beacons and asynchronous distributed key generation), randomized RSS faces circularity or expensive setups, while our deterministic RSS avoids them.

2 Prior work on RSS

We summarize the prior works in table 1 and show how they compared to ours, and elaborate on each of them below. We also summarize the solution concepts used in prior works in Appendix A.

First generation RSS. Halpern and Teague [12] consider the following preference profile for parties: (U1) rational parties' utilities only depend on the outcome of the reconstruction; (U2) each rational party prefers learning the secret than not; (U3) when one learns the secret, it prefers fewer other parties who also learn the secret. The *solution concept* utilized for solving the reconstruction game is Nash equilibrium (NE) where no party can increase their utility by unilaterally deviating from the equilibrium. They first show that the NE for the

[2] Sequential actions further allow the reconstruction to operate in *network asynchrony*. However, since our preference profile is more reasonable in network synchrony (to make "time-sensitiveness" relevant), we only discuss asynchrony setting briefly in appendix B.

Table 1: Major RSS protocols α is the probability that an instance of reconstruction is not a test run. Prior works are based on m-out-of-n secret sharing while ours is based on general access structure (see Section 3.2). The "$\sim$" sign means "in expectation". $O_\lambda(\cdot)$ means the big-O notation is hiding some security parameter λ. Coalition means private cooperation among parties. U1, U2, U3, U1*, U3* are defined in Section 4.

Protocols	Access structure	Preference profile	Non-network assumption	Network assumption	Solution concept	Side info resistant	Round complexity	Coalition resistance	Malicious parties	Communication cost blow-up
[HT04][12]	$2 \leq m < n$	U1-3	Online dealer; Trusted mediator	SBC	NE*[3]	Yes	$\sim\frac{1}{\alpha}$	1	0	$\sim O_\lambda(\frac{1}{\alpha})$
[GK06][11]	$2 \leq m \leq n$	U1-3	MPC or trusted mediator	SBC	NE*	Yes	$\sim\frac{1}{\alpha}$	$m-1$	0	$\sim O_\lambda(\frac{1}{\alpha})$
[LT06][20]	$m = \lceil n/2 \rceil$	U1-3	MPC; zero-knowledge proofs; t-NCC function [4]	SBC	ϵ-NE*	Yes	~ 2	1	$\lceil \frac{n}{2} \rceil - 2$	$\sim O_\lambda(1)$
[ADGH06][1]	$2 \leq m \leq n$	U1-2	Trusted mediator[5]	SBC	NE*	Yes	~ 2	$m-1$	0	$\sim O_\lambda(1)$
[KN08a][17]	$2 \leq m \leq n$	U1-2	Meaningful encryption; MPC	SBC [6]	Computational NE	No	$\sim\frac{1}{\alpha}$	$m-1$	0	$\sim O_\lambda(\frac{1}{\alpha})$
[KN08b][18]	$2 \leq m \leq n$	U1-2	Infinite secret domain; MPC	SBC [7]	IT strict NE	No	$\sim\frac{1}{\alpha^2}$	$m-1$	0	$\sim O(\frac{1}{\alpha}\log\frac{1}{\alpha})$
[OPRV09][25]	$2 \leq m \leq n$	U1-3	$\omega(\log n)$ honest parties	Synchronous broadcast	Bayesian NE	No	~ 2	$m-1$	0	$\sim O_\lambda(1)$
[FKN10][8]	$2 \leq m \leq n$	U1-3	VRF	Synchrony; Asynchrony	Computational strict NE	No	$\sim\frac{1}{\alpha}$	$m-1$	0	$\sim O_\lambda(1)$
[LS10][19]	$2 \leq m \leq n$	U1-3+ [8]	VRF; TDE [9]	Synchrony	Computational strict NE	Yes	$\sim\frac{1}{\alpha}$	$m-1$	0	$\sim O_\lambda(1)$
[KOTY17][14]	$2 \leq m \leq n$	U1-3	VRF authenticated SS	Synchronous broadcast	Strict NE	No	~ 3	$m-k-1$	k	$\sim O_\lambda(1)$
This work	$2 \leq m^* \leq n$ [10]	U1*, U2, U3*	None	Synchrony	SPE	Yes	2	n	$t^\star - 1$ [11]	1

[3] NE that survives elimination of weakly dominated strategies.
[4] Informally, the function is impossible to compute with $t+1$ inputs.
[5] or additional assumptions on parties' utility functions.
[6] [KN08a] achieves ϵ-NE without SBC by incurring a high multiplicative round complexity overhead that is linear in m.
[7] [KN08b] achieves ϵ-NE without SBC by incurring a high multiplicative round complexity overhead linear in the size of the secret domain.
[8] One prefers misleading others to incorrect output and not learning the secret to everyone learning the secret.
[9] Time-delayed encryption scheme constructed from memory-bound functions.
[10] m^* is the size of the smallest access group.
[11] $t^\star$ is the minimum vertex cover on the hypergraph constructed from a set of access groups: each party is a vertex; each access group is a hyperedge.

reconstruction game in a single access group is *parties not sending their shares* to others. They demonstrate that it is impossible to have a deterministic RSS protocol with a known finite number of rounds under preferences (U1)-(U3). The authors then devise a randomized RSS protocol where each round is an actual reconstruction round with probability α, and parties run the protocol until accomplishing an actual reconstruction round or after detecting deviating behaviors of not sending shares. In this way, rational parties are incentivized to broadcast shares. The RSS protocol applies to $n \geq 3$ parties (due to how the protocol realizes the probabilistic termination). It utilizes an *online dealer* who continuously issue shares, a *trusted mediator* for coordinating reconstruction, and SBC for enforcing simultaneous moves of parties. This first proposal has the following limitations:

1. An online trusted dealer is needed.
2. The protocol does not handle 2-out-of-2 secret sharing and does not have coalition resistance.
3. The protocol does not tolerate malicious parties.
4. SBC is needed.
5. The RSS designer needs to know the parties' utility functions to decide α.
6. The round complexity is $O(\alpha^{-1})$.

Second generation RSS. Gordon and Katz [11] propose a simpler RSS protocol for $n \geq 2$ parties: In each round, an honest dealer shares the actual secret which is in some field with probability α and otherwise, she shares a random element outside this field. Lysyanskaya and Triandopoulos [20] consider malicious parties and propose a scheme that tolerates $(\lceil n/2 \rceil - 1)$ malicious parties with MPC and zero-knowledge proofs. Abraham et al. [1] introduce the (k, t)-robustness notion for equilibrium strategies where a set of $k \geq 1$ parties form a coalition in playing the reconstruction game, and another set of t parties behave arbitrarily. For a base m-out-of-n Shamir's secret sharing scheme ($n \geq 2$), they propose a (k, t)-robust RSS protocol terminates in expected two rounds for $k < m \leq n - k$ and $O(\alpha^{-1})$ rounds for $k < m \leq n$ provided that parties' utility functions satisfy some additional conditions. This generation of protocols resolve issues $1, 2, 3$ and partly 6 above. However, issues 4 and 5 remain unsolved. Besides, SBC is still the main inefficiency source, on top of other newly introduced heavy tools.

Third generation RSS. Kol and Naor [18,17] remove the dependence on SBC in previous RSS protocols and adopt a stronger solution concept, strict NE[12] when solving the reconstruction game. However, [18,17] introduce high round complexity. Fuchsbauer et al. [8] utilize VRF [24] to remove the reliance on SBC and preserve the round complexity $O(\alpha^{-1})$. Moreover, the protocol applies to *asynchronous network*. However, without SBC, randomized RSS inherently does not allow for access to *side information* since an informed party can recognize a reconstruction round before sharing [3].

[12] In a strict NE, each party's equilibrium strategy generates strictly higher utility.

To accommodate side information, Lysyanskaya and Segal [19] assume computationally bounded parties and network synchrony, and utilize TDE to hide reconstruction results until the next round. Additionally, they assume parties prefer *misleading others to wrong outputs and not learning the secret* to *everyone learning the secret*. De and Pal [7] build on [19], and continue to adopt TDE for hiding shares. Side information is distributed by the dealer to help parties decide whether the reconstructed secret is correct. Knapp and Quaglia [16] build upon [7] and improve on computation overhead by employing HTLP instead of TDE. In [14], Kawachi et al. build verifiable RSS upon another RSS plus verifiable/authenticated SS (and they use [8] as the underlying RSS) – which means that they inherit all the assumptions from the underlying RSS – and achieve constant rounds in expectation. While these works relax the SBC requirement resolving issue 4, and [14] solves issue 6 (but cannot tolerate side information), the constructions are still randomized and rely on other heavy cryptographic tools or additional assumptions on rational parties' preferences.

3 Model and definitions

3.1 System and network

There are n parties functioning as share holders. They can have *side information* about the secret of interest. The parties are either **rational** and act in a utility-maximizing manner or **malicious** and behave arbitrarily. We start by assuming that the parties are rational and consider malicious parties in Section 6.

Parties are connected via authenticated point-to-point channels. Their messages are digitally signed. The network is **synchronous**, meaning that there is a known finite time bound Δ: For a message sent at some time t, it is delivered by time $t + \Delta$. We define one round to be Δ time. We assume a party learning a secret in a round x can capitalize on the information faster than a party learning it in a higher round $\geq x + 1$.

3.2 Secret sharing

This section recalls the definition of secret sharing. In this paper, we consider a more general secret sharing setting, allowing a general access structure, instead of only the threshold setting (i.e., m-out-of-n secret sharing). At a high level, it means that there exist multiple sets of participants, and each of these individual sets of participants can reconstruct the secret among themselves. All these sets together form an access structure $\mathcal{A}$. m-out-of-n is a special access structure, where each individual set is simply m out of these n participants. These are defined more formally below.

Access structure. Consider a secret sharing scheme run by n parties, $[n] = \{1, \ldots, n\}$. A general access structure $\mathcal{A} \subseteq 2^{[n]}$ is a subset of the power set of the party set. We address each set in $\mathcal{A}$ as an **access group**. The m-out-of-n threshold secret sharing scheme has access structure $\mathcal{A}^{(m)} = \{S \subseteq [n] : |S| \geq$

$m\}$. For clarity in analysis, we consider only the minimal access groups in $\mathcal{A}$ where each party in the set is needed for reconstruction and truncate their strict supersets, which we denote as $\mathcal{A}^\star$. For example, the minimal threshold access structure is $\mathcal{A}^{\star(m)} = \{S \subseteq [n] : |S| = m\}$.

The n parties can have asymmetric status in $\mathcal{A}$ in terms of the number of ways to reconstruct the secret. For instance, one party can be present in every access group and is needed by each group for secret reconstruction while this party only needs shares from any group. We capture this asymmetry by defining a new notion, *rank*, and let function $\gamma(\cdot)$ compute the rank of the input.

Definition 1 (Rank). *The initial rank of a party i in a minimal access structure $\mathcal{A}^\star$ is $\gamma(i) = |\{S \in \mathcal{A}^\star : i \in S\}|$.*

A party present in *every* access group has rank $|\mathcal{A}^\star|$, and we say that such a party is *universal* when $|\mathcal{A}^\star| > 1$. We let the lowest possible rank be 1 for non-triviality. Parties in m-out-of-n Shamir's scheme have symmetric status: They have the same rank $\binom{n-1}{m-1}$.

Next, we define a reconstruction freedom notion to measure how much a party depends on another party to reconstruct the secret.

Definition 2 (Reconstruction freedom). *For $i, j \in [n]$, the reconstruction freedom of a party j from party i is $\mathsf{free}_j(i) = |\{S \in \mathcal{A}^\star : j \in S, i \notin S\}|$.*

Secret sharing scheme. A secret sharing scheme is a tuple of two algorithms $(\mathsf{Share}(\cdot), \mathsf{Rec}(\cdot))$:

- $[\mathbf{s}] \coloneqq (\mathbf{s}_1, \ldots, \mathbf{s}_n) \leftarrow \mathsf{Share}(\mathsf{m})$: Given the secret m as input, output the shares for each party.
- $\mathsf{m}' \leftarrow \mathsf{Rec}([\mathbf{s}]_A)$ where $[\mathbf{s}]_A$ are the shares held by parties in a set A: Given a set of shares, deterministically reconstructs a secret.

The security of a secret sharing scheme requires *correctness* where any group of parties in the access structure can reconstruct the secret successfully, i.e., $\mathsf{m}' = \mathsf{m}$ if $A \in \mathcal{A}$, and *privacy* where any group of parties that does not appear in $\mathcal{A}$ do not learn anything about the secret from their received shares.

We will assume a secure secret sharing scheme defined for honest and malicious parties, and develop a RSS reconstruction algorithm that makes closed-box use of its reconstruction function $\mathsf{Rec}(\cdot)$.

3.3 Reconstruction game definitions

This section introduces concepts that are needed for our proof, arguing why our construction satisfies our requirement via game theoretic arguments.

Game representation. In game theory, a normal-form game captures the outcomes of participants playing certain strategies at the same time and receiving respective utilities. It can be formalized with (1) *the set of parties*, (2) *the actions available to each party*, and (3) *the utility function of each party*, which maps an outcome to a real number.

Our solution does not rely on SBC and therefore, allows the parties to move sequentially. Note that in the reconstruction game, sequential move does *not* mean that each party has to wait for other parties to act before taking an action but that they act in each round and take others' prior actions into consideration. When the game involves sequential moves of parties, we need to additionally capture a few more concepts, including the actions available at each point of the game, parties' knowledge of others' past actions and their own past moves, and parties' beliefs about others' future actions. Such a game is addressed and expressed as an extensive form game: It can be formalized with the three components (1)-(3) as before together with (4) the party's *knowledge of the past*, and (5) the party's *beliefs of the future* when it is her turn to move.

Solution concepts. A *strategy* is a probability distribution over all available actions, and a *strategy profile* records the strategies of all parties. Strategy profiles that have certain desired properties are called equilibrium strategies or *solution concepts*, e.g., the equilibrium where no party increases her utility by unilaterally deviating from a strategy profile is called the Nash equilibrium (NE).

Definition 3 (NE). *Let* $(\{A_i\}_{i\in[n]}, \{u_i\}_{i\in[n]})$ *denote a game where* A_i *is party* i*'s action space and* u_i *is utility function of* i*. Let* σ_i *be a probability distribution on actions in* A_i*. A vector of distributions* $\sigma = (\sigma_1, \ldots, \sigma_n)$ *is the NE if*

$$\forall i \in [n], \forall \sigma_i' \neq \sigma_i, u_i(\sigma) \geq u_i(\sigma_i', \sigma_{-i})$$

Since NE is susceptible to empty threats[13], we turn to its refinements to solve sequential games. Specifically, the solution concept that we adopt for solving the reconstruction game is subgame perfect equilibrium (SPE [26]), where the equilibrium strategy profile specifies the NE strategies for each rational party for *any* subgame.

Definition 4 (SPE). *A Nash equilibrium is said to be an SPE if and only if it is a NE in every subgame of the original game.*

Alternatively, one can consider *sequential equilirbium*: parties hold beliefs about others' past moves when selecting the utility-maximizing strategies at their turn to move, and their equilibrium strategy profile turns out to be consistent with each other's beliefs. We adopt SPE instead because parties can *observe* others' actions, e.g., sharing information or staying silent.

4 Preference profiles

This section starts with the preference profiles introduced in prior works and what we modify to obtain a new preference profile that we work on.

[13] Consider the Ultimatum game among two parties. One party proposes a way to divide a sum of money between them. If the other party accepts, then the money is divided accordingly; otherwise, both receive nothing. The responder can threaten to accept only fair offers, but this threat is not credible: given any non-zero amount, the only rational action for the responder is to accept.

Preference profiles assumed in prior works. Let $\mathsf{out}_i(r)$ denote whether party i learns the secret in a complete run r of the reconstruction game, i.e., from the start of the reconstruction algorithm until reconstruction is no longer possible even if all remaining parties are honest. $\mathsf{out}_i(r) = 1$ indicates that i learns the secret and 0 otherwise. We let $\mathsf{out}(r)$ indicate the outcome for all parties in $[n]$. We denote the utility function of party i as $u_i(\cdot)$ and i's utility from obtaining the outcome of run r as $u_i(r)$. In prior works, the assumed preference profile for any rational party $i \in [n]$ is as follows:

- (U1) For any two runs r, r', if $\mathsf{out}(r) = \mathsf{out}(r')$, then $u_i(r) = u_i(r')$.
- (U2) For any two runs r, r', if $\mathsf{out}_i(r) = 1$ and $\mathsf{out}_i(r') = 0$, then $u_i(r) > u_i(r')$.
- (U3) For any two runs r, r', if $\mathsf{out}_i(r) = \mathsf{out}_i(r')$, $\forall j \neq i, \mathsf{out}_j(r) \leq \mathsf{out}_j(r')$ and $\exists j \neq i, \mathsf{out}_j(r) < \mathsf{out}_j(r')$, then $u_i(r) > u_i(r')$.

Here U1 means that each party's utility depends only on the overall outcome. U2 means that a party strictly prefers learning the secret. U3 means that parties *strictly* prefer fewer other parties who also learn the secret.

Our preference assumption. We update the preference profile U1 and U3 to take the *timing* of learning the secret into consideration: Parties prefer to learn the secret fast[14] and *strictly* prefer fewer parties who learn the secret before themselves. Note that a party still prefers learning the secret than not learning it, and thus U2 remains the same. Also note that we adopt a weaker assumption on the timing of learning: If parties *strictly* prefer learning the secret earlier, the analysis still applies.

Let $\mathsf{it}_i(r)$ denote the earliest "round" where i learns the secret in a run r. Note that $\mathsf{it}_i(r) = \infty$ if i does not learn the secret. The updated preference profile is as follows:

- (U1*) For any two runs r, r' where $\mathsf{out}(r) = \mathsf{out}(r')$, if $\mathsf{it}_i(r) = \mathsf{it}_i(r')$, then $u_i(r) = u_i(r')$, and if $\mathsf{it}_i(r) < \mathsf{it}_i(r')$, then $u_i(r) \geq u_i(r')$.
- (U3*) For any two runs r, r' where $\mathsf{out}(r) = \mathsf{out}(r')$, if $|\{j \in [n] \wedge j \neq i : \mathsf{it}_j(r) < \mathsf{it}_i(r)\}| < |\{j \in [n] \wedge j \neq i : \mathsf{it}_j(r') < \mathsf{it}_i(r')\}|$, then $u_i(r) > u_i(r')$.

Here we do not explicitly formalize a party's preference towards letting others learn after learning the secret. However, we have assumed that parties learning the secret earlier profit from it earlier (Section 3.1). Then naturally, a party is indifferent to informing others after acting on the knowledge of the secret. In certain applications, they may find it strictly preferable to share with others, such as in the example in Section 1.

5 Fair reconstruction under the new preference profile

5.1 Order of events

To argue the security of our scheme, we need to first define a *secret reconstruction game* (see section 3.3 for a game definition).

[14] This is because we assumed that parties learning the secret in earlier rounds can profit from it faster (Section 3.1).

In the reconstruction game, the action space for each party includes: (1) abort, (2) enter a new round, and (3) send one's share(s) to one or more parties. [15] During reconstruction, one can tell if a party has taken actions (1) and (2) due to network synchrony, and action (3) directly if they are one of the recipient(s). This allows us to adopt SPE as the solution concept since rational parties can observe others' prior actions.

Given network synchrony, we can describe the reconstruction game in *rounds*. Specifically, we consider running the reconstruction algorithm among the parties only for a finite known number of rounds, $T \geq 1$. Note that if we let T be infinite, parties sending their shares or fair reconstruction can be made the equilibrium [21] with punishment on deviators. We focus on a finite known T because it is harder to encourage cooperation in this setting, which is also indicated by the impossibility result [12].

Now, consider the following **secret reconstruction game** $\mathcal{G}$ (proceeding in rounds) given a pre-defined integer $T \geq 1$.

In each round $i = 1, \ldots, T+1$:

(1) If $i = T+1$ or if each access group in $\mathcal{A}^\star$ has at least one aborted party, the game terminates.
(2) The parties in each access group with no aborted members decide whether to take one of the three actions in time Δ (after which a round ends). After each party takes an action, enter the next round and go to step (1).

One might notice that in Step (1), checking if the game has terminated may induce a computation complexity that is exponential in the number of parties due to the size of the access structure. We note that this game is conceptual and only for reasoning about parties' actions. It does not indicate the computation complexity of our actual RSS protocol (to be presented in Figure 1).

5.2 Game analysis

Now, we discuss the intuition of why under the preference profile introduced in section 4, rational parties intend to reconstruct the secret in $\mathcal{G}$. As in prior works, we aim for a fair reconstruction where all rational parties learn the secret.

Intuitions. We start with a simple case: consider an access structure with a *universal* party (who appears in every group, with rank $|\mathcal{A}^\star| > 1$). Let $u \in [n]$ be one universal party. u can ensure that it becomes the first to learn the secret by committing to only sharing information after receiving all other shares from (at least) one access group. Note that this may not be credible. After recovering the secret, u can then send all shares from one access group to other non-universal parties so that they all learn the secret after u. Then non-universal parties are better off sending their shares in the first round, since the universal party would

[15] Note that here we do not consider sending a fake share, since if the underlying base secret-sharing scheme is not verifiable, the transformed RSS scheme is not verifiable. To obtain verifiable RSS, one can simply use a verifiable secret-sharing scheme and our transformation naturally extends to the verifiable setting.

send them the shares afterwards. We later show that this strategy profile is an SPE, and the reconstruction game terminates in two rounds without requiring SBC (case (b) in Theorem 1).

Next, consider a non-singleton access structure ($|\mathcal{A}^\star| > 1$) with only non-overlapping parties (i.e., every party has rank 1). In this case, parties are incentivized to share information in the first round. This is because (1) learning the secret faster generates higher returns, (2) deviating from sharing does not improve one's utility (since others would learn at most as fast as them but not sooner), and (3) deviating from the sharing strategy may even decrease one's utility if other access groups successfully reconstruct in the first round. The *competition* among access groups who do not need each other's shares promotes cooperation in secret reconstruction inside an access group. The strategy profile of sharing information in the first round is an SPE, and the reconstruction game terminates in one round without requiring SBC. This is formalized as case (a) in Theorem 1.

Finally, consider a non-singleton access structure with only non-universal parties who have ranks $\in [1, |\mathcal{A}^\star| - 1]$, and at least one party has rank > 1. Consider a party i with rank $x > 1$, i.e., $\mathsf{rank}(i) = x > 1$. Let $A_1, \ldots, A_x$ be the x access groups that i is in, and $A = \cup_{j=1}^{x} A_j$. If for any party $j \neq i$ in set A, $\mathsf{free}_j(i) = 0$. Then i is a locally universal party for parties in set A. The reasoning in the first case described in the beginning applies: i sends everyone else shares after receiving all the other shares from at least one group. If there exists a party j such that $\mathsf{free}_j(i) > 0$, i now has incentives to share in the first round because otherwise, other access groups may reconstruct before it. This is formalized as case (c) in Theorem 1. Note that this includes m-out-of-n threshold secret sharing for all $m < n$.

Solve for SPE. We now formally state and prove the main results for RSS under preference profile U1*, U2, and U3*.

Theorem 1. *Consider a secret sharing scheme with minimal access structure $\mathcal{A}^\star$ on party set $[n]$ where $|\mathcal{A}^\star| > 1$. Suppose rational parties have preference profiles U1*, U2, and U3*, and the reconstruction algorithm is run for at most a finite T rounds for some finite integer $T > 1$.*

(a) If $\forall i \in [n], \gamma(i) = 1$, there exists a deterministic RSS scheme that terminates in 1 round.
(b) If $\exists i \in [n]$ such that $\gamma(i) = |\mathcal{A}^\star|$, there exists a deterministic RSS scheme that terminates in at most 2 rounds.
(c) In other cases, there exists a deterministic RSS scheme that terminates in at most 2 rounds.

Proof. We use backward induction to solve for the SPE. For clarity, let party i receive utility ω^i from learning the secret, lose ϵ_k^i when k parties learn the secret before themselves, and lose τ_j^i if one learns the secret in the j-th round. We have $0 = \tau_1^i \leq \tau_2^i \leq \ldots \leq \tau_T^i < \tau_{T+1}^i$ to reflect party i's desires to learn the secret sooner, and $0 = \epsilon_0^i < \epsilon_1^i < \ldots < \epsilon_{n-1}^i$ to express its preference of learning earlier

than others. For example, if party i learns the secret only before one other party in the second round, its utility equals $(\omega^i + \epsilon_1^i - \tau_2^i)$.

Setting (a): We first show that in setting (a), the SPE in the reconstruction game is parties sending their shares to each other in the first round. The equilibrium strategy of sharing information in the first round gives each party utility ω. Without loss of generality, consider an access group A with m parties $\{1, \ldots, m\}$. Consider any party $i \in A$.

- First, in the last round T, sharing gives any party $i \in A$ utility $(\omega^i + \epsilon_0^i - \tau_T^i)$ in the worst case where other access groups have already learned the secret and $(\omega^i + \epsilon_{n-m}^i - \tau_T^i)$ in the best case where no other access groups have successfully reconstructed. More generally, suppose x parties have already learned the secret in prior rounds, this number is $(\omega^i + \epsilon_{n-m-x}^i - \tau_T^i)$. All parties sharing in round T is a NE of the remaining game since no party can increase their utility by deviating towards the action of entering the next round without sharing. Each party not sharing or any $\leq (m-2)$ parties sharing are also NEs since no party can increase their utility unilaterally, which is $\epsilon_0^i - \tau_{T+1}^i$. However, the latter NEs will be eliminated when we reason backwards.
- In round $(T-1)$, all parties sharing gives each party $i \in A$ utility $(\omega^i + \epsilon_0^i - \tau_{T-1}^i)$ in the worst case and $(\omega^i + \epsilon_{n-m}^i - \tau_{T-1}^i)$ in the best case. All parties sharing is an NE of the remaining game in all cases of $0 \leq x \leq (n-m)$ other parties learn. This is because $(\omega^i + \epsilon_{n-m-x}^i - \tau_{T-1}^i) \geq (\omega^i + \epsilon_{n-m-x}^i - \tau_T^i) > (\epsilon_0^i - \tau_{T+1}^i)$.

Intuitively, if all parties are going to share in the last round or $\leq m-2$ parties share, then sharing in the second to last round provides higher utility. We apply this reasoning until round 1. Sharing in round 1 is a NE of the remaining subgame since $\omega^i > \omega^i - \tau_1^i$ for each party $i \in A$. We can apply this reasoning in each non-overlapping access group. This then concludes the SPE for setting (a). We give the RSS protocol for setting (a) in Figure 1.

Setting (b): We next show that in the setting described in (b), the SPE is as follows: In the first round, non-universal parties send their shares to universal parties, and universal parties send shares among each other; in the second round, universal parties send collected shares to non-universal parties that have sent them shares in round one. First, consider the case where there is exactly one universal party. WLOG, let this party be party 1. Consider any access group A with m parties $\{1, \ldots, m\}$.

We have established that after learning the secret, sharing is at least a weakly dominant strategy for the universal party since we assume parties learning the secret earlier can capitalize on it earlier (discussed by the end of Section 4). Next, the universal party can adopt the following strategy: after learning the secret, share the collected shares with parties who have sent shares to it in the next round. This is credible if $\tau_{T+1}^1 - \tau_1^1 < \epsilon_{n-1}^1$, meaning that learning before all others produce more premiums than the penalty from learning very late. We will discuss when this is the case because otherwise, we are back to setting (a).

- In round T, all parties in A (including party 1) sharing is an NE for the subgame if party 1 has not yet learned the secret, where each non-universal party i earns utility $(\omega^i + \epsilon^i_{n-m-x} - \tau^i_T)$ when $0 \leq x \leq n-m$ parties in $B = [n] \backslash A$ have learned the secret. Any $\leq m-3$ parties sharing are also NEs for the remainder of the reconstruction game, where each non-universal party earns $(\epsilon^i_0 - \tau^i_{T+1})$.
- In round $(T-1)$, non-universal parties in A sending shares to the universal party is the NE as they could learn before others in round T.
- In round 1, no party has learned the secret yet. If non-universal parties in A send shares to the universal party, they learn the secret in round 2. This produces higher utility than learning in later rounds 3, 4 and so on.

Next, we consider the scenario where there are multiple universal parties. WLOG, we let there be $k > 1$ universal parties and denote them as $1, \ldots, k$. Consider any access group A with m parties $\{1, \ldots, k, \ldots, m\}$, and we can apply the above reasoning in the same way. Then the equilibrium strategy profiles take at most two rounds. This concludes the SPE for setting (b). We give the RSS protocol for setting (b) in Figure 1.

Setting (c): We finally analyze setting (c). We say a party is *locally universal* if it has rank > 1, and all other parties in all its access groups are not present in any of the other access groups, i.e., other parties have to rely on this party's share in reconstruction. Combining the analysis for settings (a) and (b), the SPE in scenario (c) is as follows: For access groups with locally universal parties, in the first round, parties that are not locally universal send their shares to the locally universal parties, who send shares among themselves in the first round and send their shares to the non-locally universal parties in the second round; For access groups without universal parties, parties send their shares to their group members in the first round.

5.3 RSS protocols

Let the secret message m of length ℓ be in space $\{0,1\}^\ell$. Consider any base secret sharing scheme $(\mathsf{Share}(\cdot), \mathsf{Rec}(\cdot))$ with respect to minimal access structure $\mathcal{A}^\star$. In the sharing phase, the dealer generates shares $[\mathbf{s}] \leftarrow \mathsf{Share}(\mathsf{m})$ and distributes shares $\mathbf{s}_i$ to each party $i \in [n]$. We give the simple reconstruction routines under settings (a)-(c) in Figure 1.

Communication and computation complexity. In all three settings, the protocol terminates in at most two rounds. In the worst case, each party needs to send its share to all other parties, resulting in quadratic communication complexity. The computation complexity is bounded by the reconstruction algorithm $\mathsf{Rec}(\cdot)$ of the underlying secret sharing algorithm.

Reconstruction for setting (a)

1. Each party i multicasts (i.e., sends a message to one or more parties) $\mathbf{s}_i$ to all parties in their access groups. Wait for Δ time (i.e., one round):
   - If i collects all the shares $\mathbf{s}_A$ for its access group, denoted as $A \in \mathcal{A}^\star$, i reconstructs to obtain $\mathsf{m} \leftarrow \mathsf{Rec}(\mathbf{s}_\mathsf{A})$ and terminates.
   - Otherwise, i aborts.

Reconstruction for setting (b)

1. Each non-universal party i multicasts $\mathbf{s}_i$ to to all parties in their access groups. Each universal party u with multicast $\mathbf{s}_u$ to all the other universal parties. Wait for Δ time (i.e., one round):
   - If a universal party u collects all the shares in any of its access group, denoted as $A \in \mathcal{A}^\star$, u reconstructs to obtain $\mathsf{m} \leftarrow \mathsf{Rec}(\mathbf{s}_\mathsf{A})$ and enters into round 2. Otherwise, u aborts.
   - Non-universal parties enter into round 2.
2. u multicasts $\mathbf{s}_A$ to parties who have sent u shares in round 1 and terminates. Non-universal parties reconstruct $\mathsf{m} \leftarrow \mathsf{Rec}(\mathbf{s}_\mathsf{A})$ and terminates.

Reconstruction for setting (c)

1. Each party i who is not locally universal multicasts $\mathbf{s}_i$ to all parties in their acccess groups. Each locally universal party u sends $\mathbf{s}_u$ to other locally universal parties present in the same access groups. Wait for Δ time (i.e., one round).
   - If a locally universal party u collects all the shares in any of the access group $A \in \mathcal{A}^\star$, u reconstructs to obtain $\mathsf{m} \leftarrow \mathsf{Rec}(\mathbf{s}_\mathsf{A})$, and enters into round 2. Otherwise, u aborts.
   - Non-locally-universal parties enter into round 2.
2. u multicasts $\mathbf{s}_A$ to parties who have sent u shares. Non-locally-universal parties reconstruct $\mathsf{m} \leftarrow \mathsf{Rec}(\mathbf{s}_\mathsf{A})$ and terminates.

Fig. 1: RSS reconstruction algorithms.

6 Extension

Coalitions. Due to the preference profile, the analysis applies to any coalition size. We only need to treat the coalition as a single party in analysis, including determining its rank and steps to take in reconstruction.

Malicious parties. A malicious party can act arbitrarily, including *rushing* (as we do not require SBC or timed primitives) and never sending their shares. For malicious parties, consider the hypergraph constructed from $\mathcal{A}$: each party is a vertex; each access group is a hyperedge. The minimum cover $t^\star$ of this graph gives the maximum number of tolerable malicious parties $t^\star - 1$. This means that when there exist universal parties in $\mathcal{A}$, we cannot tolerate even one malicious corruption. Note that, however, commonly used access structures such as the

threshold structure do not have a universal party intrinsically because of the need for robustness.

Thus, to tolerate malicious parties, we first require that all the malicious parties together do not cover all access groups. Then, we slightly modify the current reconstruction algorithm: Instead of letting parties abort after learning the secrets, we let each party eventually multi-cast the secret in round T. This then makes sure at least one rational group recovers the secret, and other rational parties also learn the secret. Hence, the analysis of the rational parties remain unchanged.

MPC. Halpern and Teague [12] and Abraham et al. [1] note that the results for secret sharing apply to secure multi-party computation assuming a trusted third party and correct inputs from rational parties. The trusted mediator can perform the computation and secret-share the result among participants.

Consider the sharing phase. We follow prior RSS protocols and focus on designing a (fair) reconstruction algorithm, as it involves multiple interacting parties. Moreover, in our deterministic protocol, we do not require the dealer to re-issue actual or fake secrets as in prior works with randomized protocols. The dealer only sends shares to parties once and no more interaction is needed afterward. However, as an interesting future direction, one can examine the sharing phase of RSS as a part of a larger protocol such as MPC where the dealer can also be a share holder.

Verifiable secret sharing (VSS). Replacing the underlying secret sharing scheme in our construction with VSS *does not* change our analysis or scheme. In the analysis, first, the action of "sending a wrong share" is equivalent to "entering the next round" without sharing due to share verifiability. Second, our scheme does not employ any cryptographic tools (unlike prior works), and our solution concept does not depend on computational assumptions on parties (i.e., our SPE holds for computationally bounded parties as well). In the protocol, our construction uses the underlying secret sharing scheme as a closed-box: what happens inside is orthogonal to our analysis.

When $|\mathcal{A}| = 1$. One condition not discussed in Theorem 1 is when there is only one access group (e.g., n-out-of-n secret sharing). In this case, no competition exists. Therefore, the proof in Section 5.2 does not intuitively work. To make our construction work, we simply modify our preference profile for the parties to *strictly* prefer learning the secrets sooner than later. In other words, changing U1* to the following (difference marked in blue): (U1**) For any two runs r, r' where $\mathsf{out}(r) = \mathsf{out}(r')$, if $\mathsf{it}_i(r) = \mathsf{it}_i(r')$, then $u_i(r) = u_i(r')$, and if $\mathsf{it}_i(r) < \mathsf{it}_i(r')$, then $u_i(r) > u_i(r')$.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing. pp. 53–62 (2006), https://doi.org/10.1145/1146381.1146393

2. Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, Ł.: Secure multiparty computations on bitcoin. Communications of the ACM **59**(4), 76–84 (2016), https://doi.org/10.1145/2896386
3. Asharov, G., Lindell, Y.: Utility dependence in correct and fair rational secret sharing. Journal of Cryptology **24**, 157–202 (2011)
4. Bentov, I., Kumaresan, R.: How to use bitcoin to design fair protocols. In: Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II 34. pp. 421–439. Springer (2014), https://rdcu.be/dcEvd
5. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: 26th Annual Symposium on Foundations of Computer Science (sfcs 1985). pp. 383–395. IEEE (1985)
6. Ciampi, M., Lu, Y., Zikas, V.: Collusion-preserving computation without a mediator. IACR Cryptol. ePrint Arch. (2020)
7. De, S.J., Pal, A.K.: Achieving correctness in fair rational secret sharing. In: International Conference on Cryptology and Network Security. pp. 139–161. Springer (2013)
8. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient rational secret sharing in standard communication networks. In: Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7. pp. 419–436. Springer (2010), https://rdcu.be/dc6ap
9. Gagol, A., Leśniak, D., Straszak, D., Świetek, M.: Aleph: Efficient atomic broadcast in asynchronous networks with byzantine nodes. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 214–228 (2019)
10. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game, or a completeness theorem for protocols with honest majority. In: Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 307–328 (2019)
11. Gordon, S.D., Katz, J.: Rational secret sharing, revisited. In: Security and Cryptography for Networks: 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006. Proceedings 5. pp. 229–241. Springer (2006), https://rdcu.be/c8dwQ
12. Halpern, J., Teague, V.: Rational secret sharing and multiparty computation. In: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing. pp. 623–632 (2004), https://doi.org/10.1145/1007352.1007447
13. Katz, J.: Bridging game theory and cryptography: Recent results and future directions. In: Theory of Cryptography Conference. pp. 251–272. Springer (2008)
14. Kawachi, A., Okamoto, Y., Tanaka, K., Yasunaga, K.: General Constructions of Rational Secret Sharing with Expected Constant-Round Reconstruction. The Computer Journal **60**(5), 711–728 (12 2016). https://doi.org/10.1093/comjnl/bxw094, https://doi.org/10.1093/comjnl/bxw094
15. Kawachi, A., Okamoto, Y., Tanaka, K., Yasunaga, K.: General constructions of rational secret sharing with expected constant-round reconstruction. The Computer Journal **60**(5), 711–728 (2017)
16. Knapp, J., Quaglia, E.A.: Fair and sound secret sharing from homomorphic timelock puzzles. In: Provable and Practical Security: 14th International Conference, ProvSec 2020, Singapore, November 29–December 1, 2020, Proceedings 14. pp. 341–360. Springer (2020)
17. Kol, G., Naor, M.: Cryptography and game theory: Designing protocols for exchanging information. In: Theory of Cryptography: Fifth Theory of Cryptography

Conference, TCC 2008, New York, USA, March 19-21, 2008. Proceedings 5. pp. 320–339. Springer (2008)
18. Kol, G., Naor, M.: Games for exchanging information. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 423–432 (2008)
19. Lysyanskaya, A., Segal, A.: Rational secret sharing with side information in point-to-point networks via time-delayed encryption. Cryptology ePrint Archive (2010)
20. Lysyanskaya, A., Triandopoulos, N.: Rationality and adversarial behavior in multi-party computation. In: Advances in Cryptology-CRYPTO 2006: 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006. Proceedings 26. pp. 180–197. Springer (2006), https://rdcu.be/c8dwB
21. Mailath, G.J., Samuelson, L.: Repeated games and reputations: long-run relationships. Oxford university press (2006)
22. Malavolta, G., Thyagarajan, S.A.K.: Homomorphic time-lock puzzles and applications. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I. pp. 620–649. Springer (2019)
23. Maleka, S., Shareef, A., Rangan, C.P.: Rational secret sharing with repeated games. Lecture Notes in Computer Science **4991**, 334–346 (2008), https://link.springer.com/content/pdf/10.1007/978-3-540-79104-1.pdf#page=345
24. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: 40th annual symposium on foundations of computer science (cat. No. 99CB37039). pp. 120–130. IEEE (1999)
25. Ong, S.J., Parkes, D.C., Rosen, A., Vadhan, S.: Fairness with an honest minority and a rational majority. In: Theory of Cryptography: 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings 6. pp. 36–53. Springer (2009)
26. Selten, R.: Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics (H. 2), 301–324 (1965), https://www.jstor.org/stable/40748884
27. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979), https://doi.org/10.1145/359168.359176
28. Stadler, M.: Publicly verifiable secret sharing. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 190–199. Springer (1996)

A Compare solution concepts

ϵ-NE. NE can be relaxed to ϵ-NE to admit more equilibria.

Definition 5 (ϵ-NE). *Let $(\{A_i\}_{i\in[n]}, \{u_i\}_{i\in[n]})$ denote a game and σ_i denote a probability distribution over actions in A_i. Given a small constant $\epsilon > 0$, a vector of distributions $\sigma = (\sigma_1, \ldots, \sigma_n)$ is an ϵ-NE if*

$$\forall i \in [n], \forall \sigma_i' \neq \sigma_i, u_i(\sigma) \geq u_i(\sigma_i', \sigma_{-i}) + \epsilon$$

Computational NE. In cryptographic protocols, parties are typically computationally bounded, allowing relaxing NE to computationally bounded parties.

Definition 6 (Computational NE). *Let $(\{A_i\}_{i\in[n]}, \{u_i\}_{i\in[n]})$ denote a game and σ_i denote a probability distribution over actions in A_i. Given a security parameter κ and probablistic polynomial time players, a vector of distributions $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a computational NE if*

$$\forall i \in [n], \forall \sigma_i' \neq \sigma_i, u_i(\sigma) \geq u_i(\sigma_i', \sigma_{-i}) + \mathsf{negl}(\kappa)$$

where $\mathsf{negl}(\cdot)$ is a negligible function of the input.

Strict NE. When the NE strategy profile specifies the strictly dominant strategy for each party, it is called a strict NE. When the guarantee is information theoretic, it is called information-theoretic strict NE, and computation strict NE is defined accordingly.

Trembling-hand NE. The idea behind trembling hand NE is to accommodate errors on other players' side: even when other players make a small mistake when playing the equilibrium strategy, a player's best response is still to play the equilibrium strategy. The error can also originate from "out-of-band" event such as network failures. A proper definition hinges on defining the small error tolerance in specific applications.

Definition 7 (Trembling-hand NE [13]). *Let $(\{A_i\}_{i\in[n]}, \{u_i\}_{i\in[n]})$ denote a game, σ_i denote a probability distribution on actions in A_i for each i, and $\delta(\cdot,\cdot)$ measure the distance between two input distributions (e.g., statistical distance). Let $\epsilon > 0$ be a small constant. A vector of distributions $\sigma = (\sigma_1, \ldots, \sigma_n)$ is the stable NE with respect to ϵ trembling if*

$$\forall i \in [n], \forall \sigma_i' \neq \sigma_i', \forall \sigma_{-i}' \text{ such that } \delta(\sigma_{-i}, \sigma_{-i}') < \epsilon, u_i(\sigma_i, \sigma_{-i}') \geq u_i(\sigma_i', \sigma_{-i}')$$

Bayesian NE. Ong et al. [25] consider uncertainty of other players' types, i.e., rational or honest, and model the reconstruction game as a sequential game with incomplete information.

Definition 8 (Bayesian NE). *Let $(\{A_i\}_{i\in[n]}, \{u_i\}_{i\in[n]})$ denote a game, σ_i denote a probability distribution over actions in A_i, and μ denote the distribution of player types. A vector of distributions $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a Bayesian NE if*

$$\forall i \in [n], \forall \sigma_i' \neq \sigma_i, u_i(\mu, \sigma) \geq u_i(\mu, (\sigma_i', \sigma_{-i}))$$

B Discussion on Network Asynchrony

Now we discuss the case when the network is asynchronous, i.e., there is no finite bound on the message delay. In this case, the round notion in section 3 is no longer well-defined, since Δ does not exist anymore. Therefore, we need to deal with two issues: (1) we need to redefine our preference profile to remove the reliance on Δ while still capturing the relative order of events; (2) we need to modify our algorithm to remove the reliance on Δ.

Redefine "round". First, we redefine "round" to be the steps of actions instead of being Δ time. In other words, round i for participant j starts when j sends the i-th message out and ends when it sends the $(i+1)$-th message out or terminates. With this modifications, our preference file remains unchanged, since it relies only on rounds instead of Δ directly.

Change of the protocol in Figure 1. First, when a party takes the action of not sharing but directly "entering the next round", we require the party to send a *dummy* message. Next, to modify our scheme, we use a similar strategy as [8]: Instead of waiting for Δ time, a party waits until they obtain enough information to take the next action. For example, in setting (a), each party i waits for all the shares $\mathbf{s}_A$ in its access group. If the shares are received, it reconstructs the secret and terminates; if a dummy message is received, it aborts. This does not change our analysis.

Note that if we consider malicious parties who can always stay silent and do not send any dummy messages, we assume they do not cover all access groups for the same reason discussed in Section 6. This means that there exists at least one access group consisting of only rational parties. Rational parties are not incentivized to enter a new round without signaling as this increases their round number without facilitating other parties' next action (because it is indistinguishable from delayed messaging in other parties' perspectives).

Incompatibility with our motivation. While our scheme can indeed to be modified to fit the asynchrony setting as above, the asynchrony setting itself does not suit our motivation very well. In particular, with the "round" defined above, it is hard to motivate that one may preferring learning in one round than the other, since the second round of one party can be later than the third round of the other party in terms of global time. However, with asynchrony, it is hard to define "round" with respect to the global time directly. Therefore, asynchrony deviates from our original motivation of time-sensitive secrets. Of course, even with this definition, it still makes sense that one prefers learning in their first round rather than their second round. However, due to the subtlety above, we only discuss asynchrony in the appendix for completeness.

C Simple Solution with A Public Bulletin Board

Lastly, we discuss a simple solution to RSS that is orthogonal to our main direction (i.e., having a new preference profile). This solution is less generic

than our main direction, but adopts the preference profile proposed in [12], and thus may be of independent interest.

This simple solution is to use a public blockchain system with payment execution functions. Similar to previous works [6,4,2] for ensuring correct executions in secure computations, we can devise a straightforward solution to the RSS problem using such a system. (i) Each party first makes a sufficient deposit. (ii) The dealer then generates and distributes shares to each party according to any *publicly verifiable secret sharing (PVSS)* scheme [28]. (iii) During reconstruction, each party sends shares to the blockchain system. The deposits of parties not sharing correct information are taken to be distributed to others. As long as the deposit is sufficiently large, rational parties are incentivized to share correct information. This solution inherits the assumptions of the underlying blockchain system. Since it costs to post to blockchains, running a larger protocol with many secret sharing instances can become uneconomical. Further, our generic solution *does not* require PVSS but works with *any* secret sharing schemes.