

On the impossibility of surviving (iterated) deletion of weakly dominated strategies in rational MPC

Johannes Blömer

Jan Bobolz

Henrik Bröcher

December 23, 2022

Department of Computer Science

Paderborn University, Germany

{[bloemer](mailto:bloemer@uni-paderborn.de), [jan.bobolz](mailto:jan.bobolz@uni-paderborn.de), [henrik.broeche](mailto:henrik.broeche@uni-paderborn.de)}@uni-paderborn.de

Abstract

Rational multiparty computation (rational MPC) provides a framework for analyzing MPC protocols through the lens of game theory. One way to judge whether an MPC protocol is *rational* is through weak domination: Rational players would not adhere to an MPC protocol if deviating never decreases their utility, but sometimes increases it.

Secret reconstruction protocols are of particular importance in this setting because they represent the last phase of most (rational) MPC protocols. We show that most secret reconstruction protocols from the literature are not, in fact, rationally sound with respect to weak domination. Furthermore, we formally prove that (under certain assumptions) it is impossible to design a rationally sound secret reconstruction protocol if (1) shares are authenticated or (2) half of all players may form a coalition.

Keywords: Game Theory, Rational Secret Sharing, Multiparty Computation, Rational Cryptography, Iterated Deletion of Weakly Dominated Strategies.

1 Introduction

A multiparty computation (MPC) protocol is one that allows n parties, each with their own secret input x_i , to jointly compute the value of a function $f(x_1, \dots, x_n)$. Applications range from jointly evaluating statistics on confidential data in a privacy-preserving way to replacing trusted parties which setup cryptographic systems to substituting trusted hardware by software. Security typically ensures the correctness of results while guaranteeing to leak no more information about the inputs than the computation's result itself leaks. Traditionally, these properties must hold with respect to adversaries that are allowed to corrupt certain parties while non-corrupted parties honestly follow the protocol prescriptions.

In this paper, we are interested in *rational MPC* [HT04], i.e. rather than partitioning the MPC protocol participants into a set of strictly honest and a set of arbitrarily malicious parties, we instead analyze the parties' behavior from a game-theoretic point of view. This means that we assume that *every* participant is *rational* (rather than honest or malicious) and that they are trying to maximize some utility function. Rational MPC addresses the following issues with the standard

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" under the project number 160364472 – SFB 901/3.

MPC definition: On one hand, the standard definition is too strong because it covers arbitrarily irrational destructive behavior. On the other hand, the standard definition is too weak because it assumes that at least one party honestly executes the protocol even if it is potentially irrational to do so. Rational MPC offers an alternative that takes game-theoretic incentives into account when evaluating MPC protocols. It is the better formalization for scenarios where one can reasonably assume participants to act rationally (e.g., in economics).

Using game theory terminology, the n MPC parties are *players*. Each player i chooses a *strategy* M_i , which is an interactive Turing machine describing how they want to behave in the protocol. Then the Turing machines run their programs, interacting with each other. At the end, the utility of each player is determined, roughly speaking, by their machine’s output. An MPC protocol (M_1, \dots, M_n) is a tuple of suggested strategies for the n players, also called a *mechanism*. Following [HT04, ADGH06], we say that the protocol is a *practical mechanism* if (1) the strategies form a Nash equilibrium, i.e. no player can (significantly) improve their expected utility by deviating from M_i , and (2) the strategies survive iterated deletion of weakly dominated strategies (iDoWDS). A strategy M_i is weakly dominated if there exists an alternative strategy M_i^* that does (significantly) better against *some* strategy profile of the other players, and does *not* perform (significantly) worse against *any* strategy profile. Surviving iDoWDS means that in a process where, repeatedly, all weakly dominated strategies are deleted, the original strategy M_i is never deleted. Without properties (1) and (2), rational players have an incentive to deviate from the prescribed protocol, which is what we are trying to prevent.

Typically, rational MPC protocols work in two stages: first, the parties run a standard MPC protocol with malicious security for the functionality f . As the result of that protocol, the parties receive secret shares s_i of the computation result $s = f(x_1, \dots, x_n)$. In the second stage, the parties run a *rational* MPC secret reconstruction phase, to which each party contributes their share s_i , and the protocol yields the final result s for everyone. This structure is reminiscent of standard MPC protocols (e.g., GMW [GMW87]), which also yield a secret-sharing of the result s and then have the parties reconstruct it. In contrast to the standard setting, where secret reconstruction amounts to simply having all the (honest) parties broadcast their shares to everyone, secret reconstruction in the rational setting is much more complicated. This is because, in some sense, it is irrational for a party i to simply broadcast their share s_i [HT04]. Broadcasting the share does not help player i to reconstruct the secret, but it may help others. So for players that prefer to learn the result and prefer others not to learn the result, the simple “everyone broadcast their shares” protocol breaks down.

As a consequence, secret reconstruction protocols play a crucial role in rational MPC. The secret reconstruction scenario can be described as follows: A dealer generates a random secret s and secret shares $(s_i)_{i=1}^n$ of s , as well as digital signatures σ_i on (i, s_i) (for simplicity of exposition, we assume authentication is done via digital signatures). When using secret reconstruction as part of a larger rational MPC protocol, we can imagine that $(s_i, \sigma_i)_{i=1}^n$ are the result of some MPC computation. The player machine M_i gets as input its signed share s_i, σ_i (and the signature scheme’s public key). The machines M_1, \dots, M_n then interact with each other. Finally, each M_i outputs what it thinks the reconstructed secret is. The rational utilities that player j tries to maximize are *natural*, i.e. the player prefers outputting the correct secret over outputting a wrong secret (prefers correctness) and the player prefers other players *not* to learn the secret (prefers exclusivity).

Several works have tackled the problem of rational secret reconstruction. Halpern and Teague [HT04] have shown that no strategy that has a fixed last round can survive iDoWDS. For this reason, existing secret reconstruction protocols [HT04, ADGH06, GK06, LT06] take a randomized number of rounds, sidestepping the issue. Ultimately, parties in those protocols still broadcast shares, but there is randomness and uncertainty involved about when (non-dummy) shares are broadcast.

Another challenge for rational secret reconstruction is the authentication of the result s : If a

party i can broadcast a fake share so that all other parties receive a wrong reconstruction result $s' \neq s$ (while i can reconstruct the real result), then doing so is rational. For this reason, inherently, there needs to be some way for parties to check whether the correct share was broadcast or at least whether the reconstruction result is invalid. For this, prior work [HT04, ADGH06, GK06, LT06] has suggested using digital signatures, information-theoretically secure MACs, zero-knowledge proofs, or Reed-Solomon codes.

1.1 Our Contribution

In this paper, we show that almost all known secret reconstruction protocols, even those that sidestep the *last round* issue, still do not survive iterated deletion of weakly dominated strategies (where weak domination is adapted to the computational setting in a natural way, see Definition 10).

We observe that any “natural” strategy M_i is weakly dominated by a machine M_i^* that works as follows: M_i^* behaves exactly like M_i except that it adds an additional check to messages it receives in the first round. If *all* other players $j \neq i$ happen to send messages of the format $(\text{LEAK}, s_j, \sigma_j)$ such that σ_j is a valid signature on (j, s_j) , then M_i^* uses the received shares (s_1, \dots, s_n) to reconstruct the secret s . In this case, M_i^* continues to behave like M_i , but outputs the s from the leaked shares in the end. In all other cases, M_i^* outputs what M_i outputs.

In other words, M_i^* hopes that all other players decide to deviate from the protocol and instead simply send this special format message containing their input in plain. And indeed, if the other players play this (artificial) strategy, then M_i^* outputs the correct secret with probability 1. This is significantly better than a typical protocol M_i , which we (for now) assume just aborts because of an unexpected first-round message format (LEAK, \dots) . Furthermore, M_i^* never does worse than M_i , because the only way M_i^* deviates is by outputting a secret s that is guaranteed to be the *correct* secret (assuming unforgeable signatures), which is the preferred outcome of a rational player. So M_i^* is never worse than M_i , but does significantly better against strategies that leak their input, which means that M_i^* weakly dominates M_i . This makes intuitive sense: the additional signature check can only *help* player i , so it is irrational not to include it.

This means that any “natural” strategy M_i , which does not happen to include such a first-message check itself, is weakly dominated by the modified strategy M_i^* . It follows that M_i does not survive iterated deletion of weakly dominated strategies (iDoWDS) (or, more specifically, M_i does not even survive the first “iteration” of iDoWDS because it is weakly dominated w.r.t. the original strategy set). We show this in Section 4 using secret reconstruction protocols from the literature as examples.

It may be tempting to try to fix this issue by including the first-message check in every protocol. If M_i already checks the first message, then M_i^* does not have any advantage over M_i and does not weakly dominate it. However, there is an essentially endless supply of other ways to encode the input-leaking message. Say a strategy M_i *does* check if the first messages contain messages of the format $(\text{LEAK}, s_j, \sigma_j)$. Then this strategy is still weakly dominated by a strategy M_i^{**} , which works like M_i , but additionally checks whether the first messages have the format $(\text{LEAK}, \bar{s}_j, \bar{\sigma}_j)$, where \bar{x} denotes some other encoding, e.g., big-endian, base64 encoding, or the bitwise negation of the canonical representation. Similarly to above, M_i^{**} weakly dominates M_i , as it cannot do worse than M_i , but does better against the strategies that leak their input by sending $(\text{LEAK}, \bar{s}_j, \bar{\sigma}_j)$. Intuitively, no matter how many different ways of interpreting the first message a strategy implements, it is likely that one can come up with a new (contrived) representation that is not covered by it. So informally, it seems exceedingly unlikely that any reasonable strategy exists that survives iDoWDS. Indeed, in Section 5 we formally prove that if we allow strategies to be non-uniform Turing machines and the dealer “sufficiently” authenticates the secret shares, then there exists *no* strategy that is not weakly dominated.

What could be possible ways around this issue? For this, we examine what makes the machine M_i^* work. Because the shares are signed in the examples above, M_i^* can be sure that when it receives

authenticated first-round shares, M_i^* (almost) never outputs the wrong secret, no matter what the remaining $n - 1$ parties do. This enables the argument that M_i^* weakly dominates M_i : If it were possible for $n - 1$ parties to convince M_i^* to output a wrong secret, then M_i^* does not necessarily weakly dominate M_i anymore. So counter-intuitively, in order for the secret-sharing scenario to possess a rational mechanism (circumventing weak domination by M_i^*), the shares must not be authenticated *too well*. However, in order for a mechanism to be a Nash equilibrium, authentication must also not be *too weak*: If it were possible for a party to convince all others of a wrong secret (while receiving the correct secret himself), then doing so is rational.

There is indeed a (small) middle ground between *perfect* authentication and *no* authentication that sidesteps our initial weak domination result [ADGH06] (as we discuss below). However, in Section 6, we show that sidestepping our weak domination counterexample has its limits: If we consider *coalitions* of $n/2$ rational players, then no reasonable secret reconstruction protocol exists (at least not for typical secret-sharing schemes) that is rational to play for the coalition. Essentially, we show that in that setting, either authentication is *good enough* for the weak domination counterexample to work, or authentication is *weak enough* to enable the coalition to play a strategy that is better for them than the prescribed strategy (meaning that there is no Nash equilibrium).

1.2 Consequences

Our results call into question a wide range of rational MPC protocols, for secret reconstruction in particular. The most immediate insight is that the popular strategy of authenticating shares with digital signatures, with one-time information-theoretically secure MACs (Construction 1), or with zero-knowledge proofs seem to be widely incompatible with weak domination requirements. In all those cases, this strong authentication makes adding a first-round check to the strategy weakly dominate any reasonable protocol’s strategies.

Several secret reconstruction protocols have been proposed in the literature [ADGH06, GK06, HT04]. To sidestep the last-round issue uncovered by [HT04], they employ ideas similar to one another. All but one of those protocols exhibit the weak domination flaw we point out above. We discuss concrete examples in detail in Section 4, unifying several of the protocols in a common framework.

Note that our weak domination counterexample does not actually target the protocols themselves, but rather their setup, i.e. the authenticated shares given to the participants in the beginning. While the concrete protocols do rely on that authentication for their soundness, the underlying ideas of their protocols are still very useful. Indeed, there is one instantiation of those ideas that completely avoids our weak domination counterexample: the third protocol of Abraham, Dolev, Gonen, and Halpern [ADGH06]. Instead of authenticating the secret-sharing with signatures or MACs (as in the first two instantiations in [ADGH06]), their third instantiation uses Reed-Solomon codes (i.e. Shamir shares with redundancy) instead. This instantiation hits the sweet spot between too much and too little authentication: Reed-Solomon codes are strongly authenticating against up to $n/3$ parties (even providing error correction), but for $n - 1$ parties, it is trivial to manipulate shares to make the last party believe in a wrong secret. This allows their protocol to be a Nash equilibrium against coalitions of size at most $n/3$ while avoiding our counterexample, which requires stronger authentication. Their protocol, handling coalitions of $n/3$, is relatively close to optimal: In Section 6, we show that for coalitions of $n/2$ or larger, it is *impossible* to come up with a good protocol (under certain reasonable assumptions and definitions).

Our results also have implications for GMW-style rational MPC (e.g., [GK06, LT06]), where in the first phase, the parties run an actively secure GMW-style MPC protocol, then in the second phase, they run some rational secret reconstruction protocol to retrieve the result. Because in that scenario, the shares after the first phase are authenticated by zero-knowledge proofs, our counterexamples apply for the second phase: Our M_i^* strategy in that case would check whether in the first round

of phase two, everyone just broadcasts their shares and a zero-knowledge proof that the broadcast shares are correct, i.e. consistent to the input the parties committed to at the beginning. This strategy is, again, better against strategies that broadcast their shares in the second phase (in some format) and then halt, and it cannot do worse than the honest strategy. This is because the only value the modified strategy may output is, by the soundness of the zero-knowledge proof system, the correct computation result (which is also the only value the honest strategy would ever be willing to output).

Finally, we note that the current standard notions of rational MPC are somewhat volatile: small changes in definitions can have large effects on what mechanisms are considered “rational” to follow. Potentially, our results can be sidestepped by tweaking our definitions (which largely follow the literature, where possible), or even switching to a completely different definitional framework [KN08b, GKM⁺13, HS20].

1.3 Related Work

For the last 20 years a lot of research has been done on the interplay between game theory and cryptography (see for example the surveys [DR07, Kat08], and [MZA⁺13] for a more practically oriented perspective). This covers, at least, two different aspects: on the one hand, cryptographic approaches to game-theoretic problems, e.g. replacing mediators in certain games (see e.g. [DHR00, HNR13] and many subsequent papers); on the other hand, using game-theoretic concepts in the design of cryptographic primitives, e.g. replacing malicious adversaries by rational adversaries, or mixtures of malicious and rational adversaries (see [LT06]). The second line of research was initiated by Halpern and Teague [HT04]. Halpern and Teague initiated the study of rational multiparty computation and, in particular, rational secret sharing and secret reconstruction. Instead of designing protocols resistant to malicious adversarial behavior, they studied secret reconstruction and multiparty computation under the assumption that agents act rationally. Recently, this approach led to game-theoretic notions of fairness in multiparty coin toss and leader election [AL11, GK12, CGL⁺18, CCWS21, WAS22].

Most relevant to our work is the work of Halpern and Teague and the research that followed it [GK06, ADGH06, ACH11, KN08b, KN08a, FKN10]. In this approach, secret reconstruction, and more generally a multiparty computation of some functionality, is modeled as a game, with the goal of designing protocols that satisfy various game-theoretic properties within this game, e.g. constitute a Nash equilibrium. However, there has never been any consensus about the right definition for a good rational strategy in multiparty computation. In the rest of this paragraph, we give a brief overview of the different types of definitions for good rational strategies. Halpern and Teague originally used not only Nash equilibria but also required that a rational strategy should be stable, where they modeled stability by the well-known and above-mentioned notion of surviving *iterated deletion of weakly dominated strategies (iDoWDS)*. Whereas [GK06] followed this approach, [ADGH06] kept survival of iterated deletion of weakly dominated strategies, and strengthened Nash equilibria to coalition-resistant Nash equilibria. Kol and Naor argue that iterated deletion of weakly dominated strategies is an inadequate stability notion for rational multiparty computation, adding to purely game-theoretic criticisms raised by [Sam92] and [Sta95]. Instead, Kol and Naor proposed and used the notion of *strict* Nash equilibria. As stated by the same authors in [KN08a], strict Nash equilibria are not suited for the computational/cryptographic setting of rational multiparty computation, where one has to relax the definition of equilibria by introducing slack terms that are negligible (in a security parameter). Instead, [KN08a] consider equilibria immune to backward induction and everlasting equilibria. However, Fuchsbauer et al. [FKN10] (following suggestions in [Kat08]), define a computational notion of strict Nash equilibria and, as a stability notion, introduce stability with respect to trembles.

In this work we concentrate on iDoWDS and offer another, perhaps final, criticism of iterated

deletion of weakly dominated strategies by showing that survival of iterated deletion of weakly dominated strategies is incompatible with secret reconstruction from authenticated secret shares, as required by applications of secret sharing in more general multiparty protocols.

2 Preliminaries

2.1 Notation

In the following, define $[n] := \{1, \dots, n\}$. For index set $I \subseteq [n]$ let $-I := [n] \setminus I$, when n is clear from the context. Similarly, let $-i := -\{i\} = [n] \setminus \{i\}$ for a single index $i \in [n]$. For sets S_1, \dots, S_n , we define $S_{\times I} := \times_{i \in I} S_i$. For a vector $(s_1, \dots, s_n) \in S_{\times [n]}$, let s_I denote the restriction of s to the indices contained in I . For $s, s' \in S_{\times [n]}$, let (s_I, s'_{-I}) denote the tuple $s^* = (s_1^*, \dots, s_n^*)$ with $s_i^* := s_i$ if $i \in I$ and $s_i^* := s'_i$ otherwise. If the context is clear, we omit the additional parentheses, especially when being used within functions, e. g. we write $u(1^\lambda, s_I, s'_{-I})$ instead of $u(1^\lambda, (s_I, s'_{-I}))$. A function $\mu : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is *negligible* if $\forall c > 0 \exists \lambda_0 \forall \lambda \geq \lambda_0 : \mu(\lambda) \leq \lambda^{-c}$. A function $p : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is *noticeable* if $p(\lambda) \geq 1/q(\lambda)$ for some polynomial q .

2.2 Model of Computation and Communication

We model interactions between parties in an MPC by probabilistic polynomial-time (ppt) interactive Turing machines (ITMs), where polynomial-time corresponds to a polynomially bounded per-round running time in the security parameter λ . The security parameter is encoded as 1^λ and provided on a special tape of the interacting ITMs. These also have special tapes for incoming and outgoing communications besides the usual tapes of Turing machines. The communication proceeds in rounds, where in each round k and for each pair M_i, M_j of ITMs, M_i writes a message $m_j^{(k,i)}$, possibly the empty string, onto the outgoing communication tape for ITM M_j . At the end of a round, all messages are written onto the corresponding incoming message tapes and the next round begins. While this models simultaneous communication, which has been used for many protocols aiming to survive the iterated deletions of weakly dominated strategies (e. g. [ADGH06, GK06, LT06]), our results also transfer to models where messages may be delayed but eventually are delivered. For every ITM M_i there exists a polynomial p_i which bounds the running time for computing the outgoing messages in the security parameter. Unlike other ITM models for MPC, we make no further assumptions on the security of communication channels.

For our general result we require ITMs to be non-uniform which we define as follows.

Definition 1. A non-uniform ppt interactive Turing machine (ITM) is a pair (M, \bar{a}) where $\bar{a} = (a_1, a_2, \dots)$ is an infinite sequence of *auxiliary strings* with $|a_\lambda|$ being polynomially bounded in λ and M is a ppt ITM with a special tape for the non-uniform advice. For given input (security) parameter $\lambda \in \mathbb{N}$ and input x , M is run on $(1^\lambda, x, a_\lambda)$ where we require the running time to be polynomial in λ and the length $|x|$ of x per round of communication.

In Definition 1 we explicitly state auxiliary strings, instead of using an infinite sequence of ITMs, which facilitates the descriptions of ITMs in our theorems, examples, and proofs. Also note, if the auxiliary strings are empty, then machine M can be represented by a *uniform* ppt ITM.

2.3 Secret Sharing

A secret sharing scheme enables the owner of a secret s to share it among a set of n players P_1, \dots, P_n such that only explicitly authorized subsets of them are able to reconstruct the secret by pooling their shares. These authorized sets are defined via monotone access structures.

Definition 2 (Access Structure). Let $M = \{P_1, \dots, P_n\}$ be a set of n parties. A set \mathbb{A} of subsets of M is called *monotone* if $A \in \mathbb{A}$ and $A \subseteq B \subseteq M$ implies $B \in \mathbb{A}$. An *access structure* $\mathbb{A} \subseteq \mathcal{P}(M)$ with n parties is a *monotone collection of non-empty subsets* of M . A set $A \subseteq M$ is called *qualified* if $A \in \mathbb{A}$ and *non-qualified* if $A \notin \mathbb{A}$.

Monotonicity models that groups which are qualified to learn a shared secret remain qualified when additional parties join. In the following we define secret sharing schemes with respect to such access structures. We extend the standard secret sharing definition (c. f. [Bei11]), which only includes shares, by additional information which is used for authentication and verification of shares.

Definition 3 (Secret Sharing Scheme with locally verifiable reconstruction). Let \mathbb{A} be an access structure with n parties and \mathbb{S} be a finite set of secrets where $|\mathbb{S}| \geq 2$. A (*perfect*) *secret sharing scheme with domain of secrets \mathbb{S} realizing access structure \mathbb{A} with locally verifiable reconstruction* is a tuple of ppt algorithms $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$, where

- $\text{Setup}_\Pi(1^\lambda)$, on input security parameter 1^λ , outputs public parameters pp with $|\text{pp}| \geq \lambda$.
- $\text{Share}(\text{pp}, s)$, on input public parameters pp and secret $s \in \mathbb{S}$, outputs for each $i \in [n]$ a triple $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ consisting of share $s^{(i)}$, local verification information $\tau^{(i)} \in \{0, 1\}^*$, and authentication information $\sigma^{(i)} = (\sigma_1^{(i)}, \dots, \sigma_n^{(i)}) \in \{0, 1\}^*$.
- $\text{Recon}(\text{pp}, \tau^{(i)}, (s^{(j)}, \sigma_i^{(j)})_{j \in A})$, on input public parameters pp , P_i 's local verification information $\tau^{(i)}$, and, for $A \subseteq [n]$, tuples $(s^{(j)}, \sigma_i^{(j)})_{j \in A}$ of shares and authentication information, *deterministically* outputs an element from $\mathbb{S} \cup \{\perp\}$.

We require correctness: For all $\lambda \in \mathbb{N}$, $\text{pp} \leftarrow \text{Setup}_\Pi(1^\lambda)$, $s \in \mathbb{S}$, and for all $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})_{i \in [n]} \leftarrow \text{Share}(\text{pp}, s)$, $A \in \mathbb{A}$, $i \in A$ it holds

$$\Pr[\text{Recon}(\text{pp}, \tau^{(i)}, (s^{(j)}, \sigma_i^{(j)})_{j \in A}) = s] = 1.$$

If $\mathbb{A} = \{A \subseteq [n] \mid |A| \geq m\}$, we say Π is an *m-out-of-n* secret sharing scheme.

A secret sharing scheme with locally verifiable reconstruction is intended as follows. After the public parameters are setup, the algorithm Share is used by a dealer to share a secret s she owns. When a party P_j wants to reveal its share $s^{(j)}$ with some P_i it additionally reveals the corresponding authentication information $\sigma_i^{(j)}$. If player P_i obtained $(s^{(j)}, \sigma_i^{(j)})_{j \in A}$ corresponding to a qualified group A , by correctness Recon reconstructs the initially shared secret s using verification information $\tau^{(i)}$. Beyond these syntactical definitions, we use the following standard notion of privacy for secret sharing schemes.

Definition 4 (Perfect privacy). A secret sharing scheme Π for access structure \mathbb{A} and secret domain \mathbb{S} has *perfect privacy* if $\forall \lambda \in \mathbb{N}$, $\forall \text{pp} \leftarrow \text{Setup}_\Pi(1^\lambda)$, $\forall A \notin \mathbb{A}$, and $\forall s, s' \in \mathbb{S}$, it holds that $\text{Share}(\text{pp}, s)_A$ and $\text{Share}(\text{pp}, s')_A$ are identically distributed.

In addition to privacy we define the non-standard property of (non-uniform) local t -verifiability. Intuitively, this property ensures that it is infeasible for ppt adversaries to make an honest player output a wrong secret by manipulating up to t shares.

In $\text{Forge}_{\mathcal{A}, \Pi}^{S, C}(\lambda)$ first the secret sharing scheme is set up according to Setup_Π . Then, a secret s^* and a corresponding sharing is sampled. The adversary \mathcal{A} is given the inputs of all (corrupted) parties $i \in C$ and has to output (possibly) new shares and authentication information for these. \mathcal{A} wins if it makes one of the (non-corrupted) parties $i \in [n] \setminus C$ output a wrong secret with respect to the newly constructed values and (some of) the remaining honest values.

Experiment $\text{Forge}_{\mathcal{A},\Pi}^{\mathcal{S},C}(\lambda)$:
1. $s^* \leftarrow \mathcal{S}(1^\lambda), \text{pp} \leftarrow \text{Setup}_\Pi(1^\lambda)$.
2. $((s^{(1)}, \tau^{(1)}, \sigma^{(1)}), \dots, (s^{(n)}, \tau^{(n)}, \sigma^{(n)})) \leftarrow \text{Share}(\text{pp}, s^*)$.
3. Non-uniform adversary $(\mathcal{A}, (\omega_1, \omega_2, \dots))$ is given pp , and triples $(s^{(j)}, \tau^{(j)}, \sigma^{(j)})_{j \in C}$, and outputs $(\bar{s}^{(j)}, \bar{\sigma}^{(j)})_{j \in C}$.
4. Output is 1 iff $\exists i \in [n] \setminus C, \exists H \subseteq [n] \setminus C$ with $\text{Recon}(\text{pp}, \tau^{(i)}, (\bar{s}^{(j)}, \bar{\sigma}^{(j)})_{j \in C}, (s^{(j)}, \sigma^{(j)})_{j \in H}) \notin \{s^*, \perp\}$.

Figure 1: Experiment for local verification of secrets for secret sharing scheme Π with respect to non-uniform adversary $(\mathcal{A}, (\omega_1, \omega_2, \dots))$, set $C \subset [n]$ of corrupted parties, and family of secret distributions \mathcal{S} .

Definition 5 ((Non-uniform) local t -verifiability). A secret sharing scheme Π has *local verifiability against up to t corruptions* if \forall non-uniform ppt $\mathcal{A}, \forall C \subset [n], |C| \leq t$, there is a negligible function μ such that

$$\Pr[\text{Forge}_{\mathcal{A},\Pi}^{\mathcal{S},C}(\lambda) = 1] \leq \mu(\lambda),$$

where the experiment $\text{Forge}_{\mathcal{A},\Pi}^{\mathcal{S},C}(\lambda)$ is defined in Figure 1.

If winning experiment $\text{Forge}_{\mathcal{A},\Pi}^{\mathcal{S},C}(\lambda)$ is infeasible for any (non-uniform) ppt adversary \mathcal{A} and coalition of size t , then the scheme has (non-uniform) local verifiability against up to t corruptions (Definition 5). In particular, local $(n - 1)$ -verifiability implies that a coalition of $n - 1$ parties cannot make the remaining party P_i , using its verification information $\tau^{(i)}$, output an incorrect secret.

Note that local verifiability is different from the stronger notions of robust secret sharing (RSS) and verifiable secret sharing (VSS) (c.f. [Rab94]). In VSS the dealer may be corrupted which we do not require as we assume honest initial sharings as inputs. RSS guarantees that with respect to t deviations the initially shared secret s^* is correctly reconstructed which does not allow for throwing an error \perp . Besides these differences, we additionally choose to make the included information for authentication and verification explicit. In the following Construction 1 we give an example for a secret sharing scheme by Abraham et al. [ADGH06] which satisfies locally $(n - 1)$ -verifiable reconstruction. It essentially authenticates shares from Shamir's secret sharing scheme [Sha79] with the idea of information checking from [RB89].

Construction 1 (Secret Sharing Scheme Π^{ADGH} [ADGH06]). The m -out-of- n secret sharing scheme $\Pi^{\text{ADGH}} = (\text{Setup}^{\text{ADGH}}, \text{Share}^{\text{ADGH}}, \text{Recon}^{\text{ADGH}})$ with domain of secrets \mathbb{S} is defined as follows

- $\text{Setup}^{\text{ADGH}}(1^\lambda)$: Generates and returns the description of a field \mathbb{F} with $|\mathbb{F}| > 2^\lambda$ and $\mathbb{S} \subset \mathbb{F}$ as public parameters pp .
- $\text{Share}^{\text{ADGH}}(\text{pp}, s)$: Generates uniformly at random a degree- $(m - 1)$ polynomial $h \in \mathbb{F}[X]$ constrained by $h(0) = s$. For each $i, j \in [n], i \neq j$, it chooses uniformly at random P_i 's verification information $y_j^{(i)} \leftarrow \mathbb{F}$ and computes P_j 's corresponding authentication information $b_i^{(j)}, c_i^{(j)} \in \mathbb{F}$ such that $c_i^{(j)} = b_i^{(j)} \cdot h(i) + y_j^{(i)}$. For each $i \in [n]$, it sets $s^{(i)} = (i, h(i))$, $\tau^{(i)} = (y_1^{(i)}, \dots, y_n^{(i)})$, and $\sigma^{(i)} = ((b_1^{(i)}, c_1^{(i)}), \dots, (b_n^{(i)}, c_n^{(i)}))$, and returns $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})$.

- $\text{Recon}^{\text{ADGH}}(\text{pp}, y^{(i)}, ((j, s^{(j)}), (b_i^{(j)}, c_i^{(j)}))_{j \in A})$: Compute set of indices of valid shares as $H = \{j \in A \mid c_i^{(j)} = b_i^{(j)} \cdot s^{(j)} + y_j^{(i)}\}$. If $|H| < m$ output \perp . Otherwise choose m values $(j, s^{(j)})$, interpolate the corresponding degree- $(m-1)$ polynomial $h \in \mathbb{F}[X]$, and output $h(0)$.

2.4 Game-Theoretic Notions

In the following we define the game-theoretic notions necessary to model rationality of participants in a computational setting. These definitions mainly originate from the survey of Katz [Kat08] but are suitably adapted to our (non-uniform) setting. We begin with the definition of normal form games which provide a very basic idea for our upcoming considerations.

Definition 6 (Normal Form Game). A *normal form game* $\Gamma = ((A_i)_{i \in [n]}, (u_i)_{i \in [n]})$ with n players P_1, \dots, P_n consists of

- A set of A_i of *actions*, also called *strategies*, for each player P_i .
- A *utility function* $u_i: A_{\times[n]} \rightarrow \mathbb{R}$ for each player P_i .

We call $a \in A_{\times[n]}$ a *pure strategy profile* and $\sigma = (\sigma_1, \dots, \sigma_n)$ where σ_i denotes a distribution over A_i a (*mixed*) *strategy profile*. By $u_i(\sigma)$ we denote the expected utility of a mixed strategy profile, i.e. $u_i(\sigma) = \mathbb{E}_{a \leftarrow \sigma}[u_i(a)]$.

The sets A_i from Definition 6 define the strategies which players are allowed to play within the game. After each player has chosen a strategy, the resulting strategy profile is valued using the utility function. This, especially, enables to compare different strategy profiles and strategies based on their utility. Note, using the expected value to assess mixed strategies (and, later on, probabilistic ITMs) is a common choice which models risk-neutral players.

In order to suit interactions in a cryptographic setting we adapt this framework, as is common, in two steps. First, we introduce a security parameter 1^λ on which the utilities depend and restrict the strategies to ITMs which are ppt (in this security parameter).

Definition 7 (Computational Game). A *computational game* $\Gamma = ((S_i)_{i \in [n]}, (u_i)_{i \in [n]})$ with n players P_1, \dots, P_n consists of

- Sets S_i of ppt ITMs with (local) output space $\mathbb{O}_i \subseteq \{0, 1\}^*$ for each player P_i .
- A utility function u_i for each player P_i which maps security parameter 1^λ and (local) ITM outputs $(o_1, \dots, o_n) \in \mathbb{O}_{\times[n]}$ to a utility in \mathbb{R} .

For a given security parameter 1^λ and strategy profile $M = (M_1, \dots, M_n)$ the utility $u_i(1^\lambda, M)$ denotes the expected utility over the distribution of outputs of ITMs (induced by their randomness).

Semantically, at the beginning of a computational game the players choose their respective ITMs. Afterwards, the security parameter is fixed and the players execute their ITMs. This ordering prevents that players choose a different ITM based on the security parameter and thereby, implicitly, use non-uniform ITMs. We explicitly allow non-uniform ITMs, if a different strategy for each security parameter is intended. Also note that mixed strategies are not incorporated within the game, because probabilistic ITMs are sufficient to represent mixed strategies. In typical applications the output space of machines can be stated very precisely. For secret reconstruction the output space is the secret domain.

In the second, final step an external trusted setup is added to the framework. Such information, for example, may contain key material of a public-key infrastructure or the shares of a secret sharing. We cover this in following definition.

Definition 8 (Typed Computational Game). A *typed computational game* $\Gamma = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i)_{i \in [n]}, (u_i)_{i \in [n]})$ with n players P_1, \dots, P_n consists of

- A set \mathbb{T}_i of types for each player P_i and a corresponding ppt-sampleable *family of (input) type distributions* $\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}$ over $\mathbb{T}_{\times [n]}$.
- A set S_i of ppt ITMs with (local) output space $\mathbb{O}_i \subseteq \{0, 1\}^*$ for each player P_i .
- A utility function u_i for each player P_i which maps security parameter λ , types $(t_1, \dots, t_n) \in \mathbb{T}_{\times [n]}$, and (local) ITM outputs $(o_1, \dots, o_n) \in \mathbb{O}_{\times [n]}$ to a utility in \mathbb{R} .

For a given security parameter λ and ITMs (M_1, \dots, M_n) , we overload notation and define the utility $u_i(1^\lambda, (M_1, \dots, M_n)) = \mathbb{E}[u_i(1^\lambda, t_1, \dots, t_n, o_1, \dots, o_n)]$, where $(t_1, \dots, t_n) \leftarrow \mathcal{D}(\lambda)$ and o_i is the output of ITM $M_i(1^\lambda, t_i)$ after interacting with all the other ITMs. For a coalition $C \subseteq [n]$ we define utility $u_C(1^\lambda, (M_1, \dots, M_n)) := \sum_{i \in C} u_i(1^\lambda, (M_1, \dots, M_n))$, where each ITM $M_i, i \in C$, is run with input $(1^\lambda, (t_i)_{i \in C})$.

In a typed computational game, first the players choose their strategies, i.e. ITMs. Afterwards, the security parameter is fixed, the types (t_1, \dots, t_n) are privately sampled by an external Dealer (in game theory often called Nature), and each t_i is (privately) written on the input tape of M_i which starts the interaction. Fixing the ITMs before sampling types is of major importance with respect to types which are based on computationally hard problems. Otherwise, for example, given any discrete logarithm instance, the (computationally unbounded) player would be able to choose a strategy which hardcodes the solution to the given instance. This even exceeds the capabilities of non-uniform ITMs whose auxiliary input may only depend on the security parameter but not concrete problem instances. Utilities in typed computational games depend on the (local) outputs *and* sampled types. They are (a-priori) computed as expected value over the sampling of types, interaction of machines and their final outputs. For a coalition C of players, we define the utility u_C as sum over the parties' individual utilities when their ITMs are run on their shared inputs. This reflects the idea that in a realistic setting parties who form a coalition split up their gains.

With respect to the framework from Definition 8 the notion of t -resilient equilibria serves as first concept to describe stable strategy profiles.

Definition 9 (t -Resilient Computational Equilibrium). For a typed computational game $\Gamma = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i)_{i \in [n]}, (u_i)_{i \in [n]})$ we call strategy profile $M = (M_1, \dots, M_n) \in S_{\times [n]}$ *t -resilient computational equilibrium* if for all $C \subseteq [n], |C| = t$, and all strategies $M'_C \in S_{\times C}$ there exists a negligible function μ such that

$$u_C(1^\lambda, M'_C, M_{-C}) \leq u_C(1^\lambda, M) + \mu(\lambda)$$

Definition 9 adapts the notion of an ϵ -Nash equilibrium to the cryptographic setting of typed computational games where parties are allowed to form coalitions. In a computational equilibrium each player P_i is at most able to increase her utility by a negligible amount μ when switching to a different strategy. Assuming that players do not care about negligible improvements, a computational equilibrium is arguably stable as nobody has an incentive to deviate.

For some scenarios the stability of t -resilient equilibria is insufficient and complementary properties are demanded. One such property relies on the *dominance* of strategies which we define for typed computational games.

Definition 10 (Dominance in Typed Computational Games). Let typed computational game $\Gamma = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i)_{i \in [n]}, (u_i)_{i \in [n]})$. For player P_i a strategy $M_i^* \in S_i$ weakly dominates $M'_i \in S_i$ if

1. “Never non-negligibly worse”: For all $M_{-i} \in S_{\times-i}$ there exists a negligible function μ such that

$$u_i(1^\lambda, M_i^*, M_{-i}) \geq u_i(1^\lambda, M_i', M_{-i}) - \mu(\lambda)$$

2. “Sometimes significantly better”: There exists a noticeable function p and an opponent strategy profile $M_{-i} \in S_{\times-i}$ such that

$$u_i(1^\lambda, M_i^*, M_{-i}) \geq u_i(1^\lambda, M_i', M_{-i}) + p(\lambda)$$

If the second condition holds for all strategies, then M_i^* strictly dominates M_i' . For each player P_i , denote the set of its strictly dominated strategies by $\text{sDOM}_i(\Gamma)$ and its weakly dominated strategies by $\text{wDOM}_i(\Gamma)$.

According to Definition 10, a strategy M_i weakly dominates another strategy M_i' if (1) M_i is *at most negligibly* worse than M_i' against any opponent ITMs and (2) M_i is *noticeably* better than M_i' against at least one choice of opponent ITMs. Similarly to how computational equilibria are defined with slack (i.e. strategies that only improve by a negligible amount do not count), this notion of dominance is also adapted to the computational setting. One receives the original purely game-theoretic notion of domination when setting $\mu = p = 0$.

The purely game-theoretic notion, without slack, is not very useful for cryptographic scenarios for the same reasons that the Nash equilibrium definition has been adapted to include slack [Kat08]. In a definition without slack, any negligible improvement would be considered. In particular, say we have a protocol that involves public-key cryptography, then intuitively, any strategy can be improved by having the machine try to randomly guess the secret key (and then use that key to break something). This improvement would be, in every sense of the word, negligible (by cryptographic security guarantees) and practically completely irrelevant. However, according to the non-slack definition, it would never be considered rational to play any strategy, since the strategy with one additional brute-force attempt would be (negligibly) better. Also, it makes intuitive sense that a rational player would be indifferent to both negligible improvements and negligible loss in utility. For those reasons, dominance is defined with slack.

In contrast to Nash equilibria, there does not seem to be a consensus on how to generalize domination to a setting that includes coalitions. We refer to Section 6 for our definition of domination with coalitions.

Remark 1 (Noticeable vs non-negligible gains). With respect to the second condition of Definition 10, we could also require a non-negligible advantage from the weakly dominating strategy instead of a noticeable one. While the following results apply to both definitions, working with noticeable functions facilitates the proofs. In particular, we do not have to argue about infinite series of security parameters for which some inequality is not satisfied. Furthermore, it is questionable whether a strategy which is *non-negligibly but not noticeably* better, should be considered dominant. Using just non-negligible gains includes cases where there exists an infinite series of security parameters on which the utility is bounded by a negligible function. If, in practice, the game is only instantiated on these security parameters, the strategy would only ever have a negligible gain.

Note that essentially, weakly dominated strategies are irrelevant for the game. No rational player would consider playing them. So conceptually, weakly dominated strategies can be safely deleted from the pool of considered strategies. Deleting strategies, however, may render *other* strategies weakly dominated, with respect to the reduced strategy sets. So with the same reasoning, those “new” weakly dominated strategies should be deleted as well. This process leads to following definition of *iterated* deletion of weakly dominated strategies.

Definition 11 (Iterated Deletion of Weakly Dominated Strategies (iDoWDS)). Let typed computational game $\Gamma^0 = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i^0)_{i \in [n]}, (u_i)_{i \in [n]})$. For all $i \in [n]$ and $j \in \mathbb{N}$ define $S_i^j := S_i^{j-1} \setminus \text{wDOM}_i(\Gamma^{j-1})$ and $\Gamma^j = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i^j)_{i \in [n]}, (u_i)_{i \in [n]})$. Then $S_{\times[n]}^\infty := \bigcap_{j=1}^\infty S_{\times[n]}^j$ is the set of strategies which survives the process of iterated deletion of weakly dominated strategies and $\Gamma^\infty = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i^\infty)_{i \in [n]}, (u_i)_{i \in [n]})$ is its corresponding game. A strategy profile $(M_1, \dots, M_n) \in S_{\times[n]}^0$ survives the iterated deletion of weakly dominated strategies if $(M_1, \dots, M_n) \in S_{\times[n]}^\infty$.

During the process of deletion of strategies, the strategy sets shrink over time. While for finite strategy sets this process eventually stops, for infinite sets we are not aware of any characterizations when this process stops. For our upcoming results, it is important to note that any weakly dominated strategy w.r.t. the original (full) strategy set is deleted in the first iteration and can never be considered rational to play according to this notion. Indeed, our results will only focus on the first iteration of iterated deletion, i.e. we generally show that strategies are weakly dominated from the start (rather than becoming weakly dominated in later iterations). This implies not surviving iDoWDS, which is the term that most of the related work is concerned with.

As noted in the literature [ADGH06], for a mechanism to be considered “practical” it has to both be a Nash equilibrium and survive iterated deletion of weakly dominated strategies. Otherwise, playing such a mechanism is arguably irrational.

3 Rational Secret Reconstruction

In this section, we define the secret reconstruction game in the spirit of [GK06]. For this, we need to define the types, allowed strategies, and utility functions.

The setting is as follows. First, a secret is shared among n players using a locally verifiable secret-sharing scheme (Definition 3). This is done by a dealer in this formalization and the shares (and any authentication data) are given to each party as a type, but we could similarly imagine the shares to be the output of an MPC protocol. As another application example, a central party might secret-share authentication information among a group of people to restrict access to some application. The goal for the parties is to reconstruct the secret. Regarding the utilities, each player has a certain gain from learning the shared secret, e.g. if the result of a computation or the access to an application is valuable. How much a player gains typically also depends on whether the other parties learn the secret as well. For example, if a player is the only one to learn a password for an online banking application, then she might transfer all available money to her own account. If others also gain access to the application, then the money possibly has to be shared with them decreasing the player’s gain.

Based on these considerations, the reconstruction of locally verifiable shared secrets by rational participants is defined as follows, as an instantiation of a typed computational game (Definition 8).

Definition 12 (Secret reconstruction game with locally verifiable reconstruction). The secret reconstruction game with family of secret distributions $\{\mathcal{S}(\lambda)\}_{\lambda \in \mathbb{N}}$ over secret domain \mathbb{S} , access structure \mathbb{A} , secret-sharing scheme $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ with locally verifiable reconstruction consists of

- Type distribution $\mathcal{D}(\lambda)$: Sample public parameters $\text{pp} \leftarrow \text{Setup}_\Pi(1^\lambda)$, secret $s \leftarrow \mathcal{S}(\lambda)$, and shares $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})_{i \in [n]} \leftarrow \text{Share}(\text{pp}, s)$. Set type $t_i := (\text{pp}, (s^{(i)}, \tau^{(i)}, \sigma^{(i)}))$.
- A set S_i of ppt ITMs with (local) output space $\mathbb{S} \cup \{\perp\}$.
- A utility function u_i for each player P_i which maps security parameter, secret $s \in \mathbb{S}$, and the parties’ outputs $(s_1, \dots, s_n) \in (\mathbb{S} \cup \{\perp\})^n$ to a utility in \mathbb{R} .

Remark 2. Unlike Definition 8 of typed computational games, we let the utility function in Definition 12 only depend on the secret itself and not the whole types. This specialization reflects typical secret reconstruction utilities which only rely on the shared secret itself and not on concrete shares, authentication data, or public parameters.

Definition 12 models a scenario where players first choose the ITMs they use for reconstructing the secret which is afterwards shared among them by an external party. The secrets are distributed according to a publicly known distribution which depends on the security parameter. This dependence is especially important if the secret’s length increases with the security parameter, e. g. when it corresponds to a secret key. Then each player runs their ITM on input $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})$, consisting of the player’s share $s^{(i)}$, local verification information $\tau^{(i)}$, and authentication information $\sigma^{(i)}$ as in Definition 3. The ITM eventually outputs a guess for the secret or an error symbol \perp . After the execution, a player’s utility depends on the shared secret and the output guesses.

While utility functions might encode anything, previous works [HT04, GK06, ADGH06, KN08a] modeled players to prefer learning the (correct) secret over not learning the secret, and to prefer the others *not* to learn the (correct) secret.

Definition 13. Let u_i be the secret reconstruction utility of player P_i from a secret reconstruction game (Definition 12). We say u_i

- *prefers correctness*, if there exists a noticeable function p such that for all $\lambda \in \mathbb{N}$, secrets $s \in \mathbb{S}$, and guesses $s^*, s' \in (\mathbb{S} \cup \{\perp\})^n$ with $s_i^* = s \neq s'_i$ we have

$$u_i(1^\lambda, s, s^*) > u_i(1^\lambda, s, s') + p(\lambda).$$

- *prefers exclusivity*, if there exists $j \neq i$ and noticeable function p such that for all $\lambda \in \mathbb{N}$, secrets $s \in \mathbb{S}$, and guesses $s^*, s' \in (\mathbb{S} \cup \{\perp\})^n$ with $s_j^* = s \neq s'_j$ and $s_{-j}^* = s'_{-j}$ we have

$$u_i(1^\lambda, s, s') > u_i(1^\lambda, s, s^*) + p(\lambda).$$

If u_i prefers both correctness and exclusivity, then we call it natural.

Definition 13 states that a player prefers correctness if her utility improves when her ITM outputs the correct secret instead of the wrong secret while the others’ guesses are fixed. Additionally, a player prefers exclusivity if her utility improves when another party outputs the wrong secret instead of the right one. As discussed in previous Remark 1 on noticeable and non-negligible gains, we again choose to require noticeable functions as improvements. This restriction of utilities, which arguably applies to many real-world applications, was used to show negative results [HT04, ADGH06, LT06, AL11] as well as to construct protocols being a computational equilibrium [HT04, GK06, ADGH06, KN08a].

Finally, we restrict the distribution of secrets to be non-trivial to rule out scenarios where ITMs are able to correctly guess the secret without any interaction: The distribution must not be concentrated too much on a single secret.

Definition 14 (Non-trivial secret distribution). A family of secret distributions $\{\mathcal{S}(\lambda)\}_{\lambda \in \mathbb{N}}$ over secret domain \mathbb{S} is called *non-trivial* if there exists a noticeable function p such that for all secrets $s \in \mathbb{S}$

$$\Pr[\mathcal{S}(\lambda) = s] < 1 - p(\lambda).$$

4 Weak domination in existing secret reconstruction protocols

In this section we describe several existing strategies from [ADGH06, GK06] which were formerly claimed to survive the iterated deletion of weakly dominated strategies in the secret reconstruction game. Contradicting these claims we construct a counterexample which weakly dominates the original strategies if the initial secret-sharing scheme is locally verifiable. This counterexample serves as blueprint for other protocols like [HT04, KN08a] and provides an intuition for our general results.

The above-mentioned protocols follow the generic pattern depicted in Figure 2. We describe this pattern using standard terminology from multiparty computation, i.e. we use an ideal functionality that has to be replaced by an appropriate protocol. Using the functionality description allows us to abstract from many irrelevant details. In accordance with the secret reconstruction game, input t_i for ITM M_i^{ADGH} includes public parameters pp and a triple $(s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ consisting of share $s^{(i)}$, local verification information $\tau^{(i)}$, and authentication information $\sigma^{(i)}$. They assume there is some fake secret $\hat{s} \in \mathbb{S}$ which is not in the support of distribution \mathcal{S} of secrets and, therefore, is distinguishable from the initially shared secret s^* . The main loop always begins with a first phase where the parties query an ideal functionality $\mathcal{F}^{\beta, \hat{s}}$ (Figure 3) using their types. Functionality $\mathcal{F}^{\beta, \hat{s}}$ first checks consistency and validity of these inputs and, if successful, returns a fresh round sharing $(\bar{s}^{(i)}, \bar{\tau}^{(i)}, \bar{\sigma}^{(i)})$ of either s^* with probability β or of \hat{s} with probability $1 - \beta$. Afterwards M_i^{ADGH} sends its round share $\bar{s}^{(i)}$ and authentication information $\bar{\sigma}_j^{(i)}$ to each M_j as well as *simultaneously* obtains a message parsed as $(\bar{s}^{(j)}, \bar{\sigma}_i^{(j)})$. M_i^{ADGH} uses its round verification information $\bar{\tau}^{(i)}$ to locally reconstruct a corresponding secret. If the reconstruction fails with error symbol \perp , M_i^{ADGH} leaves the loop and only listens to any further communication. If the reconstructed secret s^* does not equal the fake secret \hat{s} , s^* is locally output as final guess. Otherwise, the loop's next round begins. Note, the protocol makes each M_i correctly output the initially shared secret s^* in an expected number of $1/\beta$ loop runs.

ITM M_i^{ADGH} on input $t_i = (\text{pp}, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ with access to $\mathcal{F}^{\beta, \hat{s}}$ (Figure 3).

```

1 : Set flag allHonest := true
2 : while allHonest do
3 :   Send  $t_i$  to  $\mathcal{F}^{\beta, \hat{s}}$  which privately returns  $(\bar{s}^{(i)}, \bar{\tau}^{(i)}, \bar{\sigma}^{(i)})$ .
4 :   For all  $M_j, j \neq i$ : Simultaneously send  $(\bar{s}^{(i)}, \bar{\sigma}_j^{(i)})$  and obtain  $(\bar{s}^{(j)}, \bar{\sigma}_i^{(j)})$ .
5 :   Compute  $s^* = \text{Recon}(\text{pp}, \bar{\tau}^{(i)}, ((\bar{s}^{(1)}, \bar{\sigma}_i^{(1)}), \dots, (\bar{s}^{(n)}, \bar{\sigma}_i^{(n)})))$ .
6 :   if  $s^* = \perp$  then
7 :     allHonest := false
8 :   elseif  $s^* \neq \hat{s}$  then
9 :     Output  $s^*$  and terminate.
10 :  endwhile
11 : Continue listening, but send nothing anymore.
```

Figure 2: Secret reconstruction strategy generalized from several protocols of [ADGH06, GK06] using an ideal functionality $\mathcal{F}^{\beta, \hat{s}}$ (Figure 3) instead of an MPC protocol.

This protocol pattern randomizes the last round in order to overcome the problem that “send no/wrong share” weakly dominates “send share” in a fixed last round. Due to the secret-sharing’s privacy, it is indistinguishable for deviating parties whether the current round’s secret equals the initial secret s^* or the fake secret \hat{s} . When a party deviates such that she makes the reconstruction

Functionality $\mathcal{F}^{\beta, \hat{s}}$ on inputs $(\mathbf{pp}_i, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ from each ITM M_i	
1:	if $\exists i \in [n] : \text{Recon}(\mathbf{pp}, \tau^{(i)}, (s^{(j)}, \sigma_i^{(j)})_{j \in [n]}) = \perp \vee \mathbf{pp}_i \neq \mathbf{pp}_1$ then
2:	return \perp
3:	else
4:	Compute $s^* = \text{Recon}(\mathbf{pp}, \tau^{(1)}, (s^{(j)}, \sigma_i^{(j)})_{j \in [n]})$
5:	Compute $(\bar{s}^{(i)}, \bar{\tau}^{(i)}, \bar{\sigma}^{(i)})_{i \in [n]} \leftarrow \begin{cases} \text{Share}(\mathbf{pp}_\Pi, s^*), & \text{with probability } \beta \\ \text{Share}(\mathbf{pp}_\Pi, \hat{s}), & \text{with probability } 1 - \beta \end{cases}$
6:	return $(\bar{s}^{(i)}, \bar{\tau}^{(i)}, \bar{\sigma}^{(i)})$ to each party P_i

Figure 3: Functionality $\mathcal{F}^{\beta, \hat{s}}$ which, given a consistent and valid sharing of secret s^* , returns a fresh sharing of s^* with probability β and of \hat{s} with probability $1 - \beta$.

either fail with \perp or a wrong secret $s \neq \hat{s}$, the remaining parties stop the interaction. If this happens in a fake round, which with probability $1 - \beta$ is the case, this stop of interaction acts as punishment as the deviating party obtains no further information on s^* .

In order to instantiate M_i^{ADGH} such that “send no/wrong share” not weakly dominates “send share”, the secret-sharing scheme, its access structure, and the parameter β have to be chosen suitably. Depending on the given utilities, these ingredients have to be chosen such that the expected loss of making the protocol stop in a fake round outweighs the expected gain of exclusively learning the secret by deviating in a non-fake round. In short, the punishment deters active deviations which are observable by the remaining players. This, however, does not account for local deviations which are not observable. To see this, consider our counterexample $\overline{M}_i^{\text{ADGH}}$ (Figure 4) which extends strategy M_i^{ADGH} by a simple check at the end of its first loop run. Concretely, $\overline{M}_i^{\text{ADGH}}$ checks whether each other machine sent a specially formatted LEAK-message containing their share and authentication information. If these values reconstruct to a valid secret under the *initial* verification information $\tau^{(i)}$, then s^* is output. $\overline{M}_i^{\text{ADGH}}$ weakly dominates the original approach M_i^{ADGH} in certain settings as specified in following theorem.

Theorem 1. Let $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ be a secret-sharing scheme (Definition 3) with perfect privacy (Definition 4). Consider a secret reconstruction game (Definition 12) for Π , with non-trivial distribution of secrets (Definition 14) and reconstruction utilities preferring correctness (Definition 13). If Π has local $(n - 1)$ -verifiability (Definition 5), then for strategy M_i^{ADGH} (Figure 2) there exists a weakly dominating strategy $\overline{M}_i^{\text{ADGH}}$.

We sketch the proof idea of Theorem 1. For more details we refer to the analogous formal proof of our generalized non-uniform result Theorem 2. In order to weakly dominate M_i^{ADGH} (Figure 2) our constructed strategy $\overline{M}_i^{\text{ADGH}}$ (Figure 4) has to be 1) noticeably better against at least one opponent strategy but 2) never more than negligibly worse against any opponent strategy. Regarding 1), consider strategies $M'_{j \rightarrow i}$ (Figure 5) which send $(\text{LEAK}, s^{(j)}, \sigma_i^{(j)})$, i. e. a specially marked message containing the initial share and authentication information, to $\overline{M}_i^{\text{ADGH}}$ and terminate. With respect to strategies $M'_{j \rightarrow i}$ ITM $\overline{M}_i^{\text{ADGH}}$ correctly parses the incoming messages, reconstructs the initial secret, and outputs it. Because M_i^{ADGH} is not instructed to parse the specific LEAK-format, reconstruction fails, M_i^{ADGH} leaves its loop, and only listens without a correct output. As we assume correctness-preferring reconstruction utilities, which value correct outputs with a noticeably higher utility than wrong outputs, requirement 1) is satisfied. Regarding 2), in comparison to M_i^{ADGH} ITM $\overline{M}_i^{\text{ADGH}}$ may only deviate and lead to a worse utility, if the remaining $(n - 1)$ -parties sent

shares which make $\overline{M_i^{\text{ADGH}}}$ reconstruct neither the initial secret s^* nor \perp under the initial $\tau^{(i)}$. Assuming local $(n-1)$ -verifiability, this happens with negligible probability against any ppt strategy $M'_{j \rightarrow i}$. Hence, compared to M_i^{ADGH} , the expected loss of $\overline{M_i^{\text{ADGH}}}$ is at most negligible which satisfies requirement 2).

ITM $\overline{M_i^{\text{ADGH}}}$ on input $t_i = (\text{pp}, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ with access to $\mathcal{F}^{\beta, \hat{s}}$ (Figure 3).
1: Run M_i^{ADGH} on the given inputs until end of first communication round.
2: For all $j \in [n] \setminus \{i\}$: Parse message m_j from M_j as $(\text{LEAK}, s^{(j)}, \sigma_i^{(j)})$.
3: Compute $s^* = \text{Recon}(\text{pp}, \tau^{(i)}, ((s^{(1)}, \sigma_i^{(1)}), \dots, (s^{(n)}, \sigma_i^{(n)})))$.
4: if $s^* \notin \{\hat{s}, \perp\}$ then // Shares valid with respect to the initial $\tau^{(i)}$ were sent.
5: Output s^* .
6: Continue to execute M_i^{ADGH} .

Figure 4: Strategy $\overline{M_i^{\text{ADGH}}}$ which weakly dominates M_i^{ADGH} (Figure 2).

ITM $M'_{j \rightarrow i}$ on input $t_i = (\text{pp}, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$ with access to $\mathcal{F}^{\beta, \hat{s}}$
1: Send $m_j = (\text{LEAK}, s^{(j)}, \sigma_i^{(j)})$ to M_i .
2: Output \perp and terminate.

Figure 5: Strategies $M'_{j \rightarrow i}$.

In order to show where our result applies, we first explain the concrete protocol instantiations and settings of M_i^{ADGH} from [ADGH06, GK06]. In the following let k be the size of coalitions, n the number of participating players, and m the threshold of required shares for an (m, n) -threshold access structure.

1. [GK06, Section 4], k -resilient equilibrium for $k < m$: As input and within the loop, requires an arbitrary secret-sharing scheme which signs the initial shares as well as round shares. The reconstruction algorithm outputs \perp if some share fails to verify under the corresponding public key. Probability β depends on utilities.
2. [ADGH06, Proposition 1], k -resilient equilibrium for $k < m$: As input, requires a secret-sharing with signed shares (as in [GK06]) or with information-theoretic 1-time MACs as in Construction 1. Within the loop a plain Shamir secret-sharing without further reconstruction is used. Probability β depends on utilities.
3. [ADGH06, Proposition 2], k -resilient equilibrium for $k < m < n - k$: As input, requires a secret-sharing with signed shares (as in [GK06]) or with information-theoretic 1-time MACs as in Construction 1. Within the loop uses Construction 1. Probability $\beta = 1/2$ but field size \mathbb{F} of Shamir sharing depends on utilities.
4. [ADGH06, Proposition 3], k -resilient equilibrium for $k < m < n - 2k$: As input and within the loop, uses a Shamir secret-sharing without further verification or authentication information. For reconstruction, uses Reed-Solomon decoding. Probability $\beta = 1/2$ but field size \mathbb{F} of Shamir sharing depends on utilities.

As the first three results all require *initial* sharings which are locally $(n-1)$ -verifiable, Theorem 1 applies. Each of these concrete protocols is weakly dominated by strategy $\overline{M_i^{\text{ADGH}}}$. Hence, differently

than claimed, these protocols do not survive the iterated deletion of weakly dominated strategies. The fourth variant ([ADGH06, Proposition 3]) does not make use of any locally verifiable properties but relies on pure combinatorics to reconstruct the original secret. In this case our counterexample does not apply. In particular, there exists an opponent strategy M_{-i} which makes ITM $\overline{M}_i^{\text{ADGH}}$ output a wrong secret by appropriately adjusting the corresponding shares. More generally, our counterexample fails in scenarios where the remaining $n - k$ parties are able to undetectably adjust their shares in order to change the reconstructed secret.

5 Provable impossibility of iterated deletion of weakly dominated strategies

As explained in the introduction, the counterexample shown in Section 4 can be counteracted by adding the same first-round check to the honest protocol. However, informally, one can argue that there are many different checks that simply expect different encodings of the special first-round message, and not all of them can be built into a polynomial-time strategy. In this section, we show that in certain settings, local $(n - 1)$ -verifiability and iterated deletion of weakly dominated strategy (iDoWDS) are provably incompatible. We start with a non-uniform setting in Section 5.1 and then discuss other settings in Section 5.2.

5.1 Impossibility with respect to non-uniform strategies

We consider the non-uniform setting. We show that for a secret reconstruction game local $(n - 1)$ -verifiability and iterated deletion of weakly dominated strategy (iDoWDS) are incompatible, i.e. in this setting every non-uniform strategy is weakly dominated by some other non-uniform strategy. This is formalized in Theorem 2 and Corollary 1. The only restrictions we need are non-trivial distributions and correctness-preferring utilities. Recall that for trivial secret distributions, i.e. distributions that are concentrated on a single secret, secret reconstruction games are mostly vacuous.

Theorem 2. Let $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ be a secret-sharing scheme (Definition 3) with perfect privacy (Definition 4). Consider a secret reconstruction game (Definition 12) for Π , with non-uniform strategies, non-trivial distribution of secrets (Definition 14), and reconstruction utilities preferring correctness (Definition 13). Let M_i be a strategy for the secret reconstruction game, i.e. a non-uniform ppt ITM. If Π has (non-uniform) local $(n - 1)$ -verifiability (Definition 5), then there exists another strategy $(M_i^*, (\omega_1, \omega_2, \dots))$ which weakly dominates M_i (Definition 10).

Corollary 1. In the non-uniform setting there exists no strategy profile for the secret reconstruction game setting described in Theorem 2 which survives the iterated deletion of weakly dominated strategies (Definition 11).

Proof (Theorem 2). In order to prove Theorem 2, given strategy M_i , we define a new strategy $(M_i^*, (\omega_1, \omega_2, \dots))$ as in Figure 6. $(M_i^*, (\omega_1, \omega_2, \dots))$ extends M_i by an additional check whether it obtained one-time pad encryptions of the original signed shares using the non-uniform keys $\omega_\lambda = (\omega_{\lambda,1}, \dots, \omega_{\lambda,n})$. Without loss of generality we assume that the messages m_j that M_i receives from other strategies are of the same length as the advice strings $\omega_{\lambda,j}$ (which in turn have the length of shares). If this is not the case, we only consider prefixes of m_j of the appropriate length. Since $(M_i^*, (\omega_1, \omega_2, \dots))$ in its first step simulates M_i until the end of the communication round, it also needs the M_i 's advice string as additional input. To simplify notation, we do not include this in the description of $(M_i^*, (\omega_1, \omega_2, \dots))$.

To prove Theorem 2, first note that $(M_i^*, (\omega_1, \omega_2, \dots))$ (Figure 6) is ppt. Next, we show its weak dominance over M_i (Definition 10). We split the proof for computational weak dominance into

Non-uniform ITM $(M_i^*, (\omega_1, \omega_2, \dots))$, $\omega_\lambda = (\omega_{\lambda,1}, \dots, \omega_{\lambda,n})$, for given M_i .
Setup: Sample $(t_1, \dots, t_n) \leftarrow \mathcal{D}(\lambda)$, $t_i = (\text{pp}, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$, and send t_i to M_i .
1: Run M_i on the given inputs until end of first communication round.
2: For all $j \in [n] \setminus \{i\}$: On message m_j from M_j set $(s^{(j)}, \sigma_i^{(j)}) := m_j \oplus \omega_{\lambda,j}$.
3: if $s^* := \text{Recon}(\text{pp}, \tau^{(i)}, (s^{(j)}, \sigma_i^{(j)})_{j \in [n]}) \neq \perp$ then
4: Output s^* .
5: Continue to execute M_i .

Figure 6: Improved strategy $(M_i^*, (\omega_1, \omega_2, \dots))$

two Lemmas 1 and 2: On the one hand, we show that M_i^* achieves at most negligibly less utility than M_i with respect to any opponent strategy M_{-i} (Lemma 1). On the other hand, we show the existence of an opponent strategy M_{-i} that achieves noticeably higher utility than M_i (Lemma 2). Taken together, Lemmas 1 and 2 show that both requirements of computational weak dominance are satisfied which finishes the proof. \square

Lemma 1. Let (non-uniform) ITM M_i be a strategy for the secret reconstruction game for a secret sharing scheme $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ with locally $(n-1)$ -verifiable reconstruction and non-uniform strategies. Then for any opponent strategy profile M_{-i} and strategy $(M_i^*, (\omega_1, \omega_2, \dots))$ (Figure 6) there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$

$$u_i(1^\lambda, M_i, M_{-i}) \leq u_i(1^\lambda, (M_i^*, (\omega_1, \omega_2, \dots)), M_{-i}) + \mu(\lambda) \quad (1)$$

Proof. For the sake of contradiction assume that for some $(M_i^*, (\omega_1, \omega_2, \dots))$, M_{-i} , and all negligible functions μ we have

$$u_i(1^\lambda, M_i, M_{-i}) > u_i(1^\lambda, (M_i^*, (\omega_1, \omega_2, \dots)), M_{-i}) + \mu(\lambda).$$

Note that the only deviation of ITM $(M_i^*, (\omega_1, \omega_2, \dots))$ from the original strategy M_i happens within lines 2-4 (Figure 6). Since, by assumption, reconstruction utilities prefer correctness, compared to M_i this deviation only decreases utility if the secret output in line 4 is not correct. In order to decrease utility more than negligibly, entering line 4 and outputting the wrong secret has to happen with a non-negligible probability. However, in that case from $(M_i^*, (\omega_1, \omega_2, \dots))$ and M_{-i} we immediately get an adversary violating the local $(n-1)$ -verifiability property of $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ (see Definition 3). \square

Lemma 2. Let ITM M_i be a strategy for the secret reconstruction game for secret sharing scheme $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ with locally $(n-1)$ -verifiable reconstruction and with non-uniform strategies (Definition 12). If the *distribution of secrets is non-trivial* (Definition 14) and *reconstruction utilities prefer correctness*, then there exist auxiliary strings $(\omega_1, \omega_2, \dots)$, an opponent strategy M_{-i} , and a noticeable function p such that for all $\lambda \in \mathbb{N}$

$$u_i(1^\lambda, (M_i^*, (\omega_1, \omega_2, \dots)), (M_{-i}, (\omega_1, \omega_2, \dots))) \geq u_i(1^\lambda, M_i, (M_{-i}, (\omega_1, \omega_2, \dots))) + p(\lambda),$$

where each strategy in profile M_{-i} gets the same sequence of auxiliary strings.

Proof. Consider the opponent strategies $(M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots))$, $j \neq i$, described in Figure 7. Together they form the profile M_{-i} .

Non-uniform ITM $(M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots)), \omega_\lambda = (\omega_{\lambda,1}, \dots, \omega_{\lambda,n})$
Setup: Sample $(t_1, \dots, t_n) \leftarrow \mathcal{D}(\lambda)$, $t_i = (\text{pp}, s^{(i)}, \tau^{(i)}, \sigma^{(i)})$, and send t_i to M_i .
1 : Send $m_j = (s^{(j)}, \sigma_i^{(j)}) \oplus \omega_{\lambda,j}$ to M_i .
2 : Output \perp and terminate.

Figure 7: Strategies $(M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots))$

The strategies in $(M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots))$ are tailored towards $(M_i^*, (\omega_1, \omega_2, \dots))$ and simply send one-time pad encryptions of their shares to M_i . Obviously, these are not useful (or rational) strategies but are still relevant for weak domination.

In the following, to ease notation, we exclude the shares verification and authentication information which are not relevant to the argument itself. Also, to increase readability, we drop the auxiliary strings from the non-uniform ITMs $(M_i^*, (\omega_1, \omega_2, \dots))$ and $(M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots))$ when possible.

For the sake of contradiction assume that for all $(\omega_1, \omega_2, \dots)$ and all noticeable functions p

$$u_i(1^\lambda, M_i^*, (M'_{j \rightarrow i})_{j \neq i}) < u_i(1^\lambda, M_i, (M'_{j \rightarrow i})_{j \neq i}) + p(\lambda). \quad (2)$$

First note, the strategies $M'_{j \rightarrow i}$ have the fixed output \perp irrespective of M_i . Therefore, against $M'_{j \rightarrow i}$, the only difference in M_i 's utility originates from the output of M_i itself. Further, because we assume utilities which prefer correctness, any output of M_i which is not the correct secret results in noticeably less utility compared to the correct secret. By construction, M_i^* always correctly reconstructs and outputs the originally shared secret in line 4 when the remaining parties run $M'_{j \rightarrow i}$. Therefore, M_i^* achieves the optimal utility with respect to the ITMs $M'_{j \rightarrow i}$. Hence, in order to satisfy Equation (2), strategy M_i has to output the correct secret with overwhelming probability for all choices of auxiliary strings. By an averaging argument this also holds when choosing the auxiliary strings uniformly at random. Formally, there exists a negligible function μ such that

$$\Pr[s \leftarrow \mathcal{S}(\lambda), \omega_\lambda \leftarrow \{0, 1\}^{\ell(\lambda)} : M_i(\text{Share}(s) \oplus \omega_\lambda) = s] = 1 - \mu(\lambda).$$

for all $\lambda \in \mathbb{N}$. We rewrite above equation as

$$\Pr[s \leftarrow \mathcal{S}(\lambda), \omega_\lambda \leftarrow \{0, 1\}^{\ell(\lambda)} : M_i(\omega_\lambda) = s] = 1 - \mu(\lambda).$$

In particular, by the uniform choice of ω the input of M_i is stochastically independent of s but M_i still outputs s with overwhelming probability. This, however, contradicts the non-trivial distribution of secrets because there exists a noticeable function p such that for any machine M' , especially M_i , we have

$$\Pr[s \leftarrow \mathcal{S}(\lambda), \omega_\lambda \leftarrow \{0, 1\}^{\ell(\lambda)} : M'(\omega_\lambda) = s] \leq \max_{S \in \mathbb{S}} \Pr[\mathcal{S}(\lambda) = s] < 1 - p(\lambda).$$

Concretely, for the negligible function μ and noticeable function p the previous equations imply relation $p(\lambda) < \mu(\lambda)$, which for λ large enough is false. \square

5.2 Impossibility with respect to other settings

If we examine the proof above, the main challenge for proving that every strategy is weakly dominated is coming up with a first-message encoding for which we can prove that the original strategy does not check it in any way. This encoding is an XOR masking the first-round message with some bit string that is the same for all machines $M'_{j \rightarrow i}$, but to which the original strategy does not have access. In the non-uniform setting, we essentially prove that a ppt machine cannot check *all* XOR

masks, and then encode some XOR mask that is not checked in the non-uniform advice string ω of the counterexample machines $(M_i^*, (\omega_1, \omega_2, \dots)), (M'_{j \rightarrow i}, (\omega_1, \omega_2, \dots))$.

Another alternative for getting an XOR mask that it is not accessed by the original strategy M_i presents itself in the random oracle model: If the original strategy M_i is such that it never queries a random oracle (e.g., any strategy in the standard model), then in the random oracle model, M_i is weakly dominated by some random oracle model strategy M_i^* . M_i^* works as in the non-uniform example, but sources the XOR mask from the random oracle (e.g., as $H(1)||H(2)||\dots$). The first-round messages of $M'_{j \rightarrow i}$ do not convey any information about the secret at all to the original non-random-oracle strategy.

Other ways are conceivable as well. For example, assume that the dealer extends each party's type t_i by some shared random bit string ω or there is some common reference string that we know is ignored by the original machine M_i (e.g., if M_i is a subprotocol in a larger protocol).

6 Impossibility of rational mechanisms for majority coalitions

In many cases, we not only want to look at individual rational actors, but also design mechanisms that are rational to follow for *coalitions* of actors [ADGH06]. So instead of standard (computational) Nash equilibria, in the coalition setting one considers t -resilient computational equilibria (Definition 9). Even though it seems not to have been done in the literature [ADGH06], we argue that in order to properly take coalitions into account, one must also account for coalitions when considering weak domination of strategies.

In this section, we provide evidence that there *cannot* be any reasonable secret-reconstruction mechanism that for coalitions of size $t \geq n/2$ is both (1) a t -resilient computational Nash equilibrium and (2) in some sense “ t -resilient against weak domination”, i.e. there is no t -coalition strategy that is sometimes (significantly) better (against some strategy of the non-coalition members) and never (non-negligibly) worse. This seems to be true as long as the secret-sharing scheme is *verifiable-or-fully-broken* (Definition 16), which is the case for the most popular secret-sharing schemes. We formally prove impossibility for those secret-sharing schemes and *non-uniform* strategies (so that we can apply a version of Theorem 2), but the result also generalizes to the settings discussed in Section 5.2 and intuitively, as argued in the introduction, similar results should apply to any reasonable concrete protocol with uniform strategies.

Intuitively, a mechanism designer has the choice between two options regarding authentication of the secret-sharing: The first option is to make the secret-sharing scheme very well authenticated, so that $n - t$ parties *cannot* convince t honest parties of a wrong secret. But then any (t -coalition) strategy is weakly dominated similar to Section 5, as the strategy that applies a share verification check to (some encoding of) the first-round messages can be sure that if the check succeeds, it outputs the *correct* secret. The alternative option is to make the secret-sharing scheme not as well authenticated, so that a coalition of $n - t$ parties *can* convince someone of a wrong secret. But in that case, no strategy can be a $(n - t)$ -resilient Nash equilibrium because it is always better for a coalition of $n - t$ parties to deviate to convince the other parties of a wrong secret. But if a strategy is not a $(n - t)$ -resilient Nash equilibrium, then it also cannot be a t -resilient Nash equilibrium because $t \geq n - t$ for $t \geq n/2$. Overall, no matter whether authentication is chosen to be strong or weak, you get a problem with either weak domination or Nash equilibria.

To prove this, we first introduce a notion of weak domination for coalitions in Section 6.1, then go on to explain our assumption on the possible secret-sharing schemes in Section 6.2, and finally prove the impossibility result in Section 6.3.

6.1 Weak domination for coalitions

First, we generalize the notion of weakly dominated strategies to weakly dominated strategies with respect to coalitions. While definitions of Nash equilibria with respect to coalitions (Definition 9) are widely available, it seems a similar generalization for weak domination is much less standard. For Nash equilibria, it is argued that if coalitions form, they may have an incentive to deviate from the prescribed mechanism in order to improve their utility. We argue that similarly, for weak domination with coalitions, it is reasonable for a coalition to deviate from the mechanism because there is an alternative coalition strategy that is never (non-negligibly) worse than the mechanism, but is (noticeably) better against *some* strategies of the non-coalition parties. We generalize Definition 10 for coalitions as follows.

Definition 15 (Dominance with coalition C). Let typed computational game $\Gamma = (\{\mathcal{D}(\lambda)\}_{\lambda \in \mathbb{N}}, (S_i)_{i \in [n]}, (u_i)_{i \in [n]})$ and $C \subseteq [n]$. A partial strategy $M_C^* \in S_{\times C}$ *weakly dominates* $M'_C \in S_{\times C}$ with respect to coalition C if

1. “Never non-negligibly worse”: For all $M_{-C} \in S_{\times -C}$, there exists a negligible function μ such that

$$u_C(1^\lambda, M_C^*, M_{-C}) \geq u_C(1^\lambda, M'_C, M_{-C}) - \mu(\lambda)$$

2. “Sometimes significantly better”: There exists a noticeable function p and a partial opponent strategy $M_{-C} \in S_{\times -C}$

$$u_C(1^\lambda, M_C^*, M_{-C}) \geq u_C(1^\lambda, M'_C, M_{-C}) + p(\lambda)$$

where u_C is defined as in Definition 8. We say that the coalition strategy $M'_C \in S_{\times C}$ is *weakly dominated* if there is some M_C^* that weakly dominates it.

We omit a definition of iterated deletion of weakly dominated strategies with respect to coalitions (it is not actually clear what that should look like, but it also is not necessary to our argument).

Theorem 2 can be generalized to coalitions as follows.

Theorem 3. Let $\Pi = (\text{Setup}_\Pi, \text{Share}, \text{Recon})$ be a secret-sharing scheme (Definition 3) with perfect privacy (Definition 4). Consider a secret reconstruction game (Definition 12) for Π , with non-uniform strategies, non-trivial distribution of secrets (Definition 14), and reconstruction utilities preferring correctness (Definition 13).

Let $C \subset [n], t = |C|$, and let $M_C = (M_i)_{i \in C}$ be some partial non-uniform strategy profile. If Π has (non-uniform) local $n - t$ -verifiability (Definition 5), then there exists a non-uniform partial strategy profile M_C^* that weakly dominates M_C (Definition 15).

Proof (sketch). Given partial strategy profile $M_C = (M_i)_{i \in C}$, define partial strategy profile M_C^* as follows. Choose $i \in C$ arbitrarily. Then M_C^* consists of $(M_i^*, (\omega_1, \omega_2, \dots))$ and strategies $M_j, j \in C \setminus \{i\}$. The rest of the proof is as the proof for Theorem 2. \square

6.2 An assumption on the secret-sharing scheme

For the results in this section, we require the secret-sharing scheme to have a specific property. Namely, we want that for any number k of corrupted shares, it must be either (1) *infeasible* to circumvent authentication (meaning it has local k -verifiability as in Definition 5), or (2) *very easy* to circumvent authentication in the following sense: Manipulating the k corrupted shares results in a sharing of a different secret s' that is *related* to the original secret s^* (even if the k parties may not be able to reconstruct s^* from their shares). Then given the related secret s' , it must be easy to find s^* . For example, for an additive (xor) secret-sharing, the process (2) can be accomplished

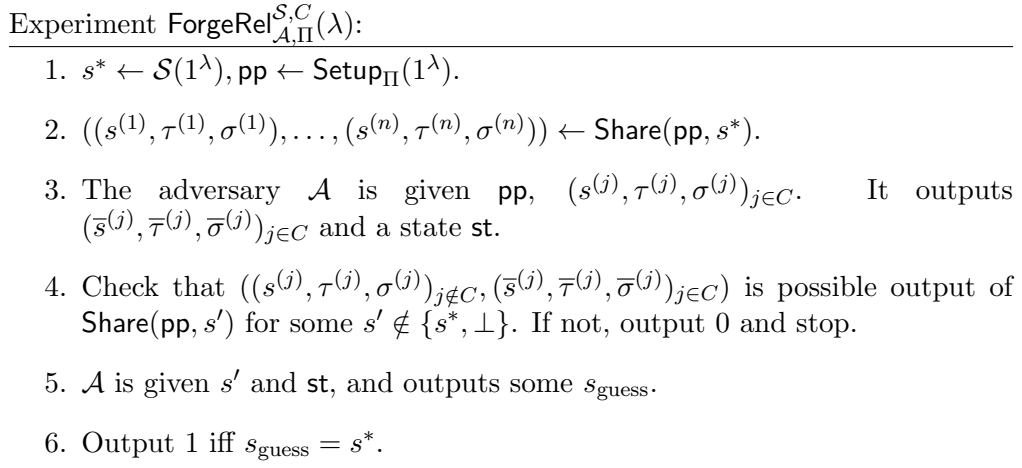


Figure 8: Experiment for fully breaking verification of secrets for secret-sharing scheme Π with respect to deterministic adversary \mathcal{A} , set $C \subset [n]$ of corrupted parties, and family of secret distributions \mathcal{S} .

by simply incrementing some corrupted share by 1, which results in a secret $s' = s^* + 1$, so given s' , it is easy to retrieve s^* . There must not be an in-between where authentication is broken against k parties, but it also is not possible for k parties to both change the sharing to a different secret and then reliably infer the real secret.

Definition 16 (Verifiable-or-fully-broken secret sharing schemes). Let Π be a secret-sharing scheme (Definition 3) for n parties. We say that Π is verifiable-or-fully-broken (for secret distributions $\mathcal{S}(1^\lambda)$) if for all $k \in [1, n - 1]$, Π has local k -verifiability, *or* there is a $C \subseteq [n], |C| = k$ and a deterministic polynomial-time algorithm \mathcal{A} such that $\Pr[\text{ForgeRel}_{\mathcal{A}, \Pi}^{\mathcal{S}, C}(\lambda) = 1] \geq 1 - \mu(\lambda)$ for some negligible function μ , where $\text{ForgeRel}_{\mathcal{A}, \Pi}^{\mathcal{S}, C}$ is as in Figure 8.

This definition covers secret-sharing schemes such as:

- Any secret-sharing scheme where shares are signed or MACed as in Construction 1 (because those schemes are k -verifiable for all $k \in [1, n - 1]$).
- Additive n -out-of- n secret-sharing (because there, \mathcal{A} can simply increment one share by $+1$ and then $s^* = s' - 1$).
- Shamir's m -out-of- n secret-sharing for $m > n/2$ with high-entropy secret distributions \mathcal{S} . This is because for $k \geq m > n - m$, \mathcal{A} can interpolate the polynomial f^* , compute all other $n - m$ parties' shares, and then set up a new polynomial f' that agrees with the $n - m$ honest parties' shares but encodes the secret $s^* + 1$; for $k < m$, the scheme is k -verifiable because if the secret has high entropy, then it is infeasible to guess the shares of the other parties. Guessing wrong results in reconstruction failing because some honest party's share does not agree with the polynomial induced by the manipulated shares output by \mathcal{A} .

These schemes are widely used and arguably the most relevant ones. Note that Shamir's secret sharing for threshold $m \leq n/2$ does not fall under this, but that case is less interesting in our setting because a coalition of $k > n/2$ can then reconstruct the secret without any interaction (in particular, if used for sharing secrets in multiparty computation, the coalition would be able to see all of it).

6.3 Proving impossibility

We are now ready to prove the following theorem.

Theorem 4. Let Π be a secret-sharing scheme (Definition 3) with perfect privacy (Definition 3) that is verifiable-or-fully-broken (Definition 16) for secret distributions \mathcal{S} . Consider the secret reconstruction game for secret sharing scheme Π with non-uniform strategies, non-trivial distribution of secrets \mathcal{S} (Definition 14), and reconstruction utilities preferring correctness and exclusivity (Definition 13). Let $t \geq n/2$. Then there exists no mechanism with the following properties:

- If everyone follows the mechanism, the correct secret is reconstructed with probability 1.
- The mechanism is a t -resilient Nash equilibrium (Definition 9).
- There is no coalition $C \subseteq [n], |C| = t$ such that M_C is weakly dominated (Definition 15).

Overall, this indicates that for most typical secret-sharing schemes, there is no pleasing mechanism that could be considered fully “rational”. In contrast to Section 5, Theorem 4 does not *assume* that the secret sharing needs to be authenticated (but rather shows that whether or not authentication is applied, both cases run into rational issues).

For the proof, there are two cases, similar to how we argued at the beginning of this section: (1) if the secret sharing scheme Π has (non-uniform) local $n - t$ -verifiability (Definition 5), then every mechanism is t -weakly dominated (because of Theorem 3). Otherwise (2) the secret sharing scheme does *not* have local $n - t$ verifiability. Then it also does not have local $t \geq n - t$ verifiability. Then Definition 16 gives us an adversary \mathcal{A} that manipulates the coalition shares, altering the shared secret from s^* to some $s' \neq s^*$ (for the non-coalition parties), and can output the correct s^* for the coalition parties. We use \mathcal{A} to construct a coalition strategy with better utility than the mechanism, meaning that the mechanism is not a t -resilient computational Nash equilibrium.

Proof (Theorem 4). The statement follows from Theorem 3 for the case that Π has (non-uniform) local $n - t$ -verifiability, and from Lemma 3 in the other case. \square

Behavior of ITM M_i^* in the coalition ($i \in C$) on input $(\text{pp}, (s^{(j)}, \tau^{(j)}, \sigma^{(j)}))_{j \in C}$
where M_i is the honest strategy and \mathcal{A} (Definition 16) fully breaks Π .
1 : Run $((\bar{s}^{(j)}, \bar{\tau}^{(j)}, \bar{\sigma}^{(j)})_{j \in C}, \text{st}) \leftarrow \mathcal{A}(\text{pp}, (s^{(j)}, \tau^{(j)}, \sigma^{(j)})_{j \in C})$.
2 : Run $M_i(\text{pp}, \bar{s}^{(i)}, \bar{\tau}^{(i)}, \bar{\sigma}^{(i)})$ interactively, until M_i outputs s' .
3 : Run $s_{\text{guess}} \leftarrow \mathcal{A}(s', \text{st})$.
4 : Output s_{guess} .

Figure 9: Improved strategy M_i^* for coalition member $i \in C$

Lemma 3. In the setting of Theorem 4, assume Π does not have (non-uniform) local t -verifiability. Then no mechanism $(M_1, \dots, M_n) \in S_{\times [n]}$ is a t -resilient computational Nash equilibrium.

Proof. Let $M = (M_1, \dots, M_n)$ be a mechanism. Let C and \mathcal{A} be as in Definition 16, $C \subseteq [n], |C| = t$. Let $(M_i^*)_{i \in C}$ be as in Figure 9.

Consider a run of strategies $((M_i^*)_{i \in C}, (M_i)_{i \notin C})$ from the point of view of the coalition strategies M_i^* . If \mathcal{A} outputs manipulated shares that are possible output of $\text{Share}(\text{pp}, s')$ for some secret s' , the result of the honestly run mechanism will be s' . This is because all the coalition members get

the same output from the deterministic \mathcal{A} , and the honestly executed mechanism always succeeds in reconstructing the input shared secret (in this case the manipulated one).

That means that from the point of view of \mathcal{A} , everything is exactly as in $\text{ForgeRel}_{\mathcal{A}, \Pi}^{S, C}(\lambda)$. So that with overwhelming probability, the coalition members output the right secret $s^* = s_{\text{guess}}$ and the non-coalition members output a wrong secret $s' \neq s^*$. Because parties prefer exclusivity, it follows that the coalition utility $\sum_{i \in C} u_i(1^\lambda, (M_i^*)_{i \in C}, M_{-C})$ with the strategies M_i^* is noticeably larger than the coalition utility $\sum_{i \in C} u_i(1^\lambda, M)$ for the mechanism (where everyone learns the correct secret). Hence M is not a t -resilient Nash equilibrium. \square

References

- [ACH11] Gilad Asharov, Ran Canetti, and Carmit Hazay. Towards a game theoretic view of secure computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 426–445. Springer, 2011.
- [ADGH06] Ittai Abraham, Danny Dolev, Rica Gonen, and Joseph Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the 25th Annual Symposium on Principles of Distributed Computing, PODC*, pages 53–62. ACM, 2006.
- [AL11] Gilad Asharov and Yehuda Lindell. Utility dependence in correct and fair rational secret sharing. *Journal of Cryptology*, 24(1):157–202, 2011.
- [Bei11] Amos Beimel. Secret-sharing schemes: A survey. In *Coding and Cryptology - Third International Workshop, IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011.
- [CCWS21] Kai-Min Chung, T-H Hubert Chan, Ting Wen, and Elaine Shi. Game-theoretic fairness meets multi-party protocols: the case of leader election. In *Annual International Cryptology Conference*, pages 3–32. Springer, 2021.
- [CGL⁺18] Kai-Min Chung, Yue Guo, Wei-Kai Lin, Rafael Pass, and Elaine Shi. Game theoretic notions of fairness in multi-party coin toss. In *Theory of Cryptography Conference*, pages 563–596. Springer, 2018.
- [DHR00] Yevgeniy Dodis, Shai Halevi, and Tal Rabin. A cryptographic solution to a game theoretic problem. In *Annual International Cryptology Conference*, pages 112–130. Springer, 2000.
- [DR07] Yevgeniy Dodis and Tal Rabin. Cryptography and game theory. *Algorithmic game theory*, pages 181–207, 2007.
- [FKN10] Georg Fuchsbauer, Jonathan Katz, and David Naccache. Efficient rational secret sharing in standard communication networks. In *Theory of Cryptography Conference*, pages 419–436. Springer, 2010.
- [GK06] S. Dov Gordon and Jonathan Katz. Rational secret sharing, revisited. In *Security and Cryptography for Networks, 5th International Conference, SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 229–241. Springer, 2006.
- [GK12] Adam Groce and Jonathan Katz. Fair computation with rational players. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 81–98. Springer, 2012.

- [GKM⁺13] Juan A. Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 648–657. IEEE Computer Society, 2013.
- [GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987*, pages 218–229. ACM, 1987.
- [HNR13] Pavel Hubáček, Jesper Buus Nielsen, and Alon Rosen. Limits on the power of cryptographic cheap talk. In *Annual Cryptology Conference*, pages 277–297. Springer, 2013.
- [HS20] John Hillas and Dov Samet. Dominance rationality: A unified approach. *Games Econ. Behav.*, 119:189–196, 2020.
- [HT04] Joseph Y. Halpern and Vanessa Teague. Rational secret sharing and multiparty computation: extended abstract. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing,*, pages 623–632. ACM, 2004.
- [Kat08] Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. In *5th Theory of Cryptography Conference, TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 251–272. Springer, 2008.
- [KN08a] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 320–339. Springer, 2008.
- [KN08b] Gillat Kol and Moni Naor. Games for exchanging information. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 423–432, 2008.
- [LT06] Anna Lysyanskaya and Nikos Triandopoulos. Rationality and adversarial behavior in multi-party computation. In *Annual International Cryptology Conference*, pages 180–197. Springer, 2006.
- [MZA⁺13] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Başçar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):1–39, 2013.
- [Rab94] Tal Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. ACM*, 41(6):1089–1109, 1994.
- [RB89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 73–85. ACM, 1989.
- [Sam92] Larry Samuelson. Dominated strategies and common knowledge. *Games and Economic Behavior*, 4(2):284–313, 1992.
- [Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
- [Sta95] Dale O Stahl. Lexicographic rationalizability and iterated admissibility. *Economics Letters*, 47(2):155–159, 1995.
- [WAS22] Ke Wu, Gilad Asharov, and Elaine Shi. A complete characterization of game-theoretically fair, multi-party coin toss. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 120–149. Springer, 2022.