Information-Theoretically Secure MPC against Mixed Dynamic Adversaries

Ivan Damgård, Daniel Escudero, Divya Ravi *

Aarhus University, J.P. Morgan AI Research**, Aarhus University,
ivan@cs.au.dk, daniel.escudero@jpmorgan.com, divya@cs.au.dk

Abstract. In this work we consider information-theoretically secure MPC against an *mixed* adversary who can corrupt t_p parties passively, t_a parties actively, and can make t_f parties fail-stop. With perfect security, it is known that every function can be computed securely if and only if $3t_a+2t_p+t_f < n$, and for statistical security the bound is $2t_a+2t_p+t_f < n$. These results say that for each given set of parameters (t_a, t_p, t_f) respecting the inequality, there exists a protocol secure against this particular choice of corruption thresholds. In this work we consider a *dynamic* adversary. Here, the goal is a *single* protocol that is secure, no matter which set of corruption thresholds (t_a, t_p, t_f) from a certain class is chosen by the adversary. A dynamic adversary can choose a corruption strategy after seeing the protocol and so is much stronger than a standard adversary.

Dynamically secure protocols have been considered before for computational security. Also the information theoretic case has been studied, but only considering non-threshold general adversaries, leading to inefficient protocols.

We consider threshold dynamic adversaries and information theoretic security. For statistical security we show that efficient dynamic secure function evaluation (SFE) is possible if and only if $2t_a + 2t_p + t_f < n$, but any dynamically secure protocol must use $\Omega(n)$ rounds, even if only fairness is required. Further, general reactive MPC is possible if we assume in addition that $2t_a + 2t_f \leq n$, but fair reactive MPC only requires $2t_a + 2t_p + t_f < n$.

For perfect security we show that both dynamic SFE and verifiable secret sharing (VSS) are impossible if we only assume $3t_a + 2t_p + t_f < n$ and remain impossible even if we also assume $t_f = 0$. On the other hand, perfect dynamic SFE with guaranteed output delivery (G.O.D.) is possible when either $t_p = 0$ or $t_a = 0$ i.e. if instead we assume $3t_a + t_f < n$ or $2t_p + t_f < n$. Further, perfect dynamic VSS with G.O.D. is possible under the additional conditions $3t_a + 3/2t_f \leq n$ or $2t_p + 2t_f \leq n$. These conditions are also sufficient for dynamic perfect reactive MPC.

* Funded by the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 803096 (SPEC)

** Work done while Daniel Escudero was at Aarhus University

1 Introduction

In secure multiparty computation (MPC) a set of n parties want to compute an agreed function on inputs held privately by the parties such that the intended result is the only new information released. We want that this holds, even if some parties are corrupted by an adversary. One may consider different types of corruption: passive corruption, where the adversary observes the state of the party as it executes the protocol, active corruption where the adversary controls the action of the party, and finally fail-stop corruption where the corrupted party is honest, but can be forced to stop the protocol prematurely.

In most of the work on MPC, it is assumed that the adversary does only passive or only active corruption. However, in [FHM98] the notion of a mixed adversary was studied, that is, one that can corrupt t_a players actively, t_p players passively, and can fail-stop corrupt t_f players. It was shown that every function can be computed securely with perfect security if and only if $3t_a + 2t_p + t_f < n$, while for statistical security the bound is $2t_a + 2t_p + t_f < n$. This was for the case of a synchronous network with secure point-to-point channels and additionally a broadcast channel in the case of statistical security (which is also the network model we use in this paper).

A mixed adversary protocol is more flexible and hence is sometimes preferable in practice: if we consider only active corruptions then for perfect security we must always assume less than $n/3$ corruptions. But if we make the realistic assumption that a large number of players might crash while only a small number of players are actively corrupted, we can tolerate faulty behavior by more than $n/3$ players. For instance, we can tolerate that t_a is about $n/9$ while t_f is about $2n/3$.

It is important to understand what these feasibility results actually mean: namely what they say is that for each given set of parameters (t_a, t_p, t_f) respecting the inequality, there exists a protocol secure against this particular choice of corruption thresholds. We will call such a choice a *corruption strategy* in the following. However, one may instead consider a fundamentally different type of adversary, known as a *dynamic* adversary. Here, the goal is to design a *single* protocol that is secure for any corruption strategy that respects the inequality. In other words, a dynamic adversary can choose a corruption strategy after seeing the protocol and so is clearly much stronger than a standard adversary.

The feasibility of dynamically secure protocols has been considered before for computational security in [HLM13], where the notion of a dynamic adversary was introduced, but where only passive and active corruptions were considered. In [PR19], the exact round complexity for MPC in this model was determined. In particular, it was shown that the number of rounds must be linear in n, in contrast to the non-dynamic case, where constant round is possible.

The case of information theoretically secure dynamic MPC has also been (indirectly) considered before, in [BFH+08] and [HMZ08], where security against general mixed adversaries was studied. A general mixed adversary may choose to actively, respectively passively, respectively fail corrupt players in three different subsets, where the triple of subsets must be chosen from a family of triples known

as an adversary structure. Since the actual triple (corruption strategy) chosen by the adversary is not given to the protocol, this model also covers the mixed dynamic adversary model we described above. In a nutshell, our model is the threshold version of the general mixed adversary model, where the adversary is limited to adversary structures described only by subset sizes t_a, t_p and t_f. In [BFH+08] and [HMZ08] combinatorial characterizations were given of those adversary structures for which one can achieve MPC with guaranteed output delivery and perfect, respectively statistical security. The protocols presented in these works all have complexity polynomial in the size of the adversary structure, i.e., in the number of subsets it contains. This means that the complexity is typically exponential in the number of players, even when restricting to the threshold case.

This state of affairs leaves open a number of important questions: First, can we achieve complexity polynomial *in the number of players* in the threshold case? Second, must the number of rounds be $\Omega(n)$ also for the case of information theoretic security? Note that the lower bound for computational security from [PR19] does not cover our case: while we consider a stronger form of security, the number of corruptions is smaller (since otherwise perfect or statistical security is not possible) and it is not clear whether this might allow for constant rounds protocols (which indeed we know exist for the non-dynamic case). Finally, while the result in [BFH+08] and [HMZ08] characterize the adversary structures for which dynamic MPC with guaranteed output delivery is possible, it might be the case that weaker security guarantees such as security with abort or fairness can still be achieved for a larger class of structures.

1.1 Our Contribution

In this work we focus on (threshold) dynamic mixed adversaries in the information theoretic setting, and we give some answers to the above open questions, which, to the best of our knowledge have not been considered before. Our primary focus is to determine feasibility conditions (in terms of thresholds (t_a, t_p, t_f)) that are necessary and sufficient for information theoretic secure function evaluation (SFE) and reactive MPC against dynamic mixed adversaries for two different security levels – namely, (a) Fairness (i.e. adversary gets the output only if honest parties do) and (b) Guaranteed Output Delivery (G.O.D i.e., the adversary cannot prevent honest parties from obtaining the output)

Along the way of addressing the above primary questions of interest, we also touched upon the following two dimensions for some classes of protocols (which we elaborate below) that give further insight about protocols secure against dynamic adversaries – (a) round complexity and (b) security with abort (weaker notion of security where the adversary may obtain the output while honest parties do not).

We elaborate on our results below – note that whenever we say that some security goal can be (efficiently) achieved, we mean that there is a protocol achieving it with complexity polynomial in the number of players.

We will be considering separately two types of functionalities. The first is *Secure Function Evaluation (SFE)*, that is, functionalities that simply receive input and deliver some function of the inputs. This should be contrasted with the stronger notion *reactive MPC*, that is, functionalities that keep state and can receive inputs and deliver outputs several times.

An important example of a reactive functionality is Verifiable Secret-Sharing (VSS), where a dealer inputs a secret value, and the functionality will later reveal it, on request from all honest players. We note for future reference that in any setting where both VSS and SFE is possible, we can also do general reactive MPC. Namely we can use the standard approach where players provide input to the reactive functionality by doing VSS. We can now compute on the inputs using an SFE that takes the VSS shares as input, it then delivers the desired outputs as well as a set of VSS shares to the players to define the new state.

To state our results, we will need the concept of a *threshold adversary structure*. Such a structure is a set $\mathcal{S}$ of corruptions strategies that the adversary can choose from, i.e., a set of triples (t_a, t_p, t_f). Note that a bound such as $2t_a + 2t_p + t_f < n$ can be thought of as shorthand for an adversary structure, namely the one containing all triples satisfying the inequality.

1.1.1 Statistical Security In a nutshell: for statistical security, we obtain tight characterisations for feasibility of dynamic SFE and dynamic reactive MPC.

In more detail: For statistical security we show that dynamically-secure SFE with G.O.D is possible for a dynamic adversary that respects $2t_a + 2t_p + t_f < n$. This completes the picture, since even non-dynamic SFE is impossible if the inequality is violated.

Considering reactive MPC, we first establish the conditions for existence of dynamic VSS. Let $\mathcal{S}$ be a threshold adversary structure. Let $R_{\mathcal{S}}$ be the maximal value of $t_a + t_p$ than can occur in $\mathcal{S}$ and let $F_{\mathcal{S}}$ be the maximal value of $t_a + t_f$. It is easy to see (and also follows from the results in [HMZ08]) that dynamic VSS with G.O.D. is impossible if $R_{\mathcal{S}} + F_{\mathcal{S}} \geq n$. We show that efficient dynamic VSS with G.O.D. is possible for any $\mathcal{S}$ that satisfies $R_{\mathcal{S}} + F_{\mathcal{S}} < n$, and also satisfies the general feasibility condition $2t_a + 2t_p + t_f < n$. An example of an adversary structure $\mathcal{S}$ that would satisfy this condition is (the set of all triples satisfying) $2t_a + 2t_p + 2t_f < n$. But other tradeoffs between the parameters are also possible.

From this we conclude that reactive MPC with G.O.D. is possible for an adversary structure $\mathcal{S}$ if and only if $R_{\mathcal{S}} + F_{\mathcal{S}} < n$ and $2t_a + 2t_p + t_f < n$ holds for all triples in $\mathcal{S}$.

On the other hand, we show that VSS with fair reconstruction only requires $2t_a + 2t_p + t_f < n$, and from this we conclude that fair reactive MPC is possible if and only if $2t_a + 2t_p + t_f < n$.

1.1.2 Perfect Security In a nutshell: for perfect security, we obtain tight characterisations for feasibility of dynamic SFE and dynamic reactive MPC with G.O.D., in the cases where either $t_a = 0$ or $t_p = 0$. We obtain a general tight characterisation for feasibility of VSS with fair reconstruction.

In more detail: For perfect security, the feasibility condition for a non-dynamic adversary is $3t_a + 2t_p + t_f < n$, this of course must be satisfied for dynamic protocols to exist. However, we show that even if we assume $t_f = 0$ (so the adversary must respect $3t_a + 2t_p < n$), dynamic SFE with security with abort is impossible, so in particular G.O.D. is also impossible (from the study in [BFH+08] one can derive a similar result that only rules out G.O.D.). It is then natural to consider what happens if we weaken the adversary in the other two ways that come to mind, namely by setting $t_p = 0$ or $t_a = 0$, so the adversary must respect $3t_a + t_f < n$ or $2t_p + t_f < n$. It turns out that in these two cases, efficient perfect G.O.D. dynamic SFE is possible. While allowing a maximal number of active, or of passive corruptions are natural cases to consider, other tradeoffs (where all three parameters can be non-zero) may also allow for efficient and perfect G.O.D. dynamic SFE (non-efficient such protocols are implied by the results in [BFH+08]). We leave the exploration of this for future work.

For reactive MPC, we derive from the results in [BFH+08] that dynamic VSS with G.O.D. is impossible assuming only the non-dynamic feasibility condition $3t_a + 2t_p + t_f < n$, and remains impossible even if we assume $t_f = 0$. Similarly to the case of SFE, we then explore what happens if we weaken the adversary in the other two natural ways, by setting $t_a = 0$ or $t_p = 0$. For the case of $t_a = 0$, we show that the condition $2t_p + 2t_f \leq n \ \wedge \ 2t_p < n$ is necessary and sufficient for dynamic G.O.D. VSS. When $t_p = 0$, we show that $3t_a + 3/2t_f \leq n \ \wedge \ 3t_a < n$ is necessary and sufficient for dynamic G.O.D. VSS.

For the parameter ranges where the positive VSS results apply, we also have SFE, so by combining the two, we conclude: when $t_a = 0$, general dynamic MPC with G.O.D. is possible if and only if $2t_p + 2t_f \leq n \ \wedge \ 2t_p < n$. when $t_p = 0$ it is possible if and only if $3t_a + 3/2t_f \leq n \ \wedge \ 3t_a < n$.

Finally, we show that for dynamic perfect VSS with fair reconstruction, the non-dynamic bound $3t_a + 2t_p + t_f < n$ is necessary and sufficient. Since the conditions for SFE are stronger, this shows that dynamic, perfect and fair reactive MPC is possible whenever dynamic perfect SFE is possible.

1.1.3 Round Complexity of Dynamic Statistical SFE

We show that, even if the protocol is only required to be fair, any dynamic statistically secure SFE protocol must use $\Omega(n)$ rounds. This shows that dynamic security comes at a price in this setting. Namely, against a non-dynamic adversary, we can have constant-round statistically (in some cases even perfectly) secure protocols for any function, if we do not demand that protocol is efficient in terms of computational complexity [IK02]. Furthermore, it is well-known that even if we do insist on computational efficiency, we can still have constant round SFE for a large class of functions.

Our dynamically secure SFE protocol completes the picture as it can be instantiated to require only $O(n)$ rounds [1].

Figure 1 shows a more concise overview of our contributions.

[1] In a bit more detail, our construction needs as subprotocol a general non-dynamic SFE protocol π, and the complexity we obtain is n times that of π. Efficient non-constant

	General non-dynamic feasibility condition	Positive results	Negative results
SFE	Statistical $2t_a + 2t_p + t_f < n$	GOD SFE	Fairness requires $\Omega(n)$ rounds
	Perfect $3t_a + 2t_p + t_f < n$	GOD SFE, if $t_a = 0$ / $t_p = 0$	SFE with abort if $t_f = 0$
Reactive MPC	Statistical $2t_a + 2t_p + t_f < n$	Fair MPC. GOD MPC if $R_{\mathcal{S}} + F_{\mathcal{S}} < n$	GOD VSS if $R_{\mathcal{S}} + F_{\mathcal{S}} \geq n$
	Perfect $3t_a + 2t_p + t_f < n$	Fair MPC whenever SFE is possible. GOD MPC if: $t_a = 0$ & $2t_p + 2t_f \leq n$ $t_p = 0$ & $3t_a + 3/2t_f \leq n$	GOD VSS if: $t_f = 0$ $t_a = 0$ & $2t_p + 2t_f > n$ $t_p = 0$ & $3t_a + 3/2t_f > n$

Fig. 1. Overview of the results presented in this paper. (t_a, t_p, t_f) refers to the thresholds for active, passive and fail-stop corruptions respectively. $R_{\mathcal{S}}$ denotes the maximal number of player states the adversary can read and $F_{\mathcal{S}}$ is the maximal number of players that can abort, where $\mathcal{S}$ is the set of corruption strategies the adversary can choose from. The positive results all assume the general feasibility condition listed in the first column, in some cases additional conditions are listed as required.

Open Questions. As mentioned above, some intriguing questions left open by our work include (but are not limited to) the following directions – (a) Exploring dimensions of round complexity (which we addressed for fair statistical SFE) and security with abort (which we addressed for perfect SFE, to strengthen our negative result) for other classes of protocols. (b) We chose to determine the additional feasibility conditions for protocols with perfect security allowing maximal active or passive corruption. However, other tradeoffs (where t_a, t_p, t_f are all non-zero) may also be possible, exploring this is left open by our work.

1.2 On Modeling of Fail-Stop Corruptions

One may consider two types of fail-stop corruptions, based on whether the adversary is allowed to see the messages that fail-stop parties would send to

round protocol π exists for all functions, so our construction is always efficient if we do not insist on asymptotically tight (but still polynomial) round complexity. However, if π is constant round we obtain $O(n)$ rounds. Such a protocol π exists for all functions but is not always computationally efficient. Of course, it would be nice if our $O(n)$ result could be shown with computational efficiency for all functions, but this would be extremely surprising: if the number of players is constant, it would imply constant-round, information theoretically secure and computationally efficient protocol for all functions. Doing this, even for a constant number of players, has been open for decades and is probably a very hard problem. On the other hand, if the function in question has an efficient non-dynamic constant-round protocol, as many functions do, then we can use that one as subprotocol and get an efficient dynamic $O(n)$-round protocol.

corrupted parties during the round where they are set to crash, or not [2]. We refer to the former as "rushing fails" and the latter as "non-rushing fails". Both models require the adversary to specify, in the beginning of the round, the identity of the parties intended to fail-stop in that round.

All our positive results hold against rushing fails (where the adversary is stronger). Conversely, all our negative results hold even for non-rushing fails, with the exception of the lower bound on the round complexity of statistically secure SFE. We leave the round complexity for non-rushing fails as an open problem.

This refinement of how fail-stop is modelled does not seem to have been considered in the literature before. Previous works consider non-rushing fails, and in particular the general mixed adversary protocols in [BFH+08] and [HMZ08] do not seem to be secure against rushing fails.

1.3 Technical overview

1.3.1 Secure function evaluation. To prove the lower bound on number of rounds for statistical security, we create a sequence of attacks that will force the protocol to use an additional round for each attack. This is inspired by Patra et al. [PR19], but we need to design a completely new set of attacks for our setting. This is because the existing result uses the interplay between passive and active corruptions, whereas we must exploit fail-stop corruptions. This makes the problem harder: for passive and active corruptions the adversary has access to the state of the corrupted parties, while this is not the case for fail-stop corruptions. The feasibility result for statistical security follows the template from Hirt et al. [HLM13]: we first run a protocol with maximal threshold that will output a set of secret-sharings. These contain additive shares of the result with different thresholds and we then open these in a carefully chosen sequence. This prevents the adversary from getting the output unfairly. Crucially, we generate shares in the output "masked" with a random value (as opposed to just the output as in previous works). The mask is given to all players, but the adversary will not learn it if he only does fail-corruptions. We need this trick to tolerate a dynamic adversary with rushing fails. If players fail or misbehave, we can eliminate them and rerun to get G.O.D.

For perfect security, the impossibility result for SFE can be obtained by a reduction to an impossibility result for 3 parties from Fitzi et al. [FHM98]. This result basically says that if the adversary can corrupt one of the first two players passively, or the third player actively, then the AND function cannot be computed securely.

1.3.2 Reactive MPC. First of all, a simple reactive functionality such as VSS does not allow secure computation per se, so lower bounds for SFE do not in general carry over to the reactive setting. Conversely, as we explain in a moment,

[2] In the case of statistical security, this includes the message that those parties were about to send on the broadcast channel, even if no one is actively or passively corrupted.

for a dynamic adversary, it is sometimes the case that SFE is possible but VSS is not. Hence, the results for reactive MPC are of a different nature.

For statistical security, impossibility of VSS with G.O.D when $R_{\mathcal{S}} + F_{\mathcal{S}} \geq n$ follows easily: Recall that $\mathcal{S}$ is the set of corruption strategies the adversary can choose from, $R_{\mathcal{S}}$ is the maximal number of player states the adversary can read and $F_{\mathcal{S}}$ is the maximal number of players that can abort. This means that in any VSS the secret must be determined from the state of the $n - F_{\mathcal{S}}$ remaining number of players. But since $R_{\mathcal{S}} \geq n - F_{\mathcal{S}}$ is the maximal value of $t_a + t_p$, this means the adversary always learns the secret.

Note that if the goal was instead SFE, it would be an option to eliminate the players who crashed and rerun the protocol, this will work as long as nothing about inputs was revealed. But this does not always work for VSS: a dynamic adversary can choose a large number of fail corruptions and only activate them after the sharing phase is over. Note that this issue is specific for dynamic protocols. A non-dynamic protocol is allowed to know that a large number of fail corruptions may happen and this will allow it to run with a smaller privacy threshold and survive the crashes.

On the other hand, if $R_{\mathcal{S}} + F_{\mathcal{S}} < n$, we show a construction of VSS protocol with G.O.D that uses our statistical SFE upper bound to realize the sharing with the appropriate threshold (to maintain privacy), followed by reconstruction which is G.O.D. due to presence of sufficient number of honest and passively corrupt parties. For the construction of VSS with fair reconstruction against dynamic adversary (with no additional assumption), we re-use the technique of secret-sharings with different thresholds.

The feasibility result for perfect fair VSS uses a modification of the technique in [BGW88] based on bi-variate polynomials to get consistent secret-sharings of the input with different thresholds, which we can then open gradually. As far as we know, bi-variate polynomials have not been used for dynamic security before. Notably, they work to create consistent secret-sharings whenever $3t_a+2t_p+t_f < n$, despite this condition being insufficient for dynamic SFE and reconstruction with guaranteed output delivery. In the setting of perfect security, we cannot rely on authentication of shares for reconstruction, so we must rely on error-correction instead. This means that the argument for fairness during the gradual opening becomes very delicate: as the adversary is dynamic, we do not know the number of errors and erasures in advance, but we still need to make sure that the error correction will always either work correctly or return an error.

Lastly, we remark that some of the techniques described above are also employed in our positive results related to the special cases of G.O.D. VSS with $t_a = 0$ and $t_p = 0$. We refer to the respective technical sections for details. The negative results for these cases are derived by translating the characterizations of [BFH+08] to the threshold case, as we describe in Section 6. Notably, this translation turned out to be non-trivial. The conditions from [BFH+08] are complicated and it is not immediate to see what they say about the threshold case. In particular, we exploit our positive result for fair VSS here, because it

shows that one of combinatorial feasibility conditions from [BFH+08] is implied already by $3t_a + 2t_p + t_f < n$ and so can be ignored in our analysis.

1.4 Related Work

As mentioned earlier, the works of [HLM13,PR19] study dynamic adversaries in the computational setting. In the information-theoretic setting, *non-dynamic* mixed adversaries (where protocols are parameterized by thresholds (t_a, t_p, t_f)) have been studied in various works such as [FHM98,HLMR11,HM20].

As described earlier, information theoretic secure SFE and MPC against general mixed adversaries was studied in [BFH+08,HMZ08]. Combinatorial characterizations were given of the adversary structures that allow for SFE and reactive MPC, with perfect security in [BFH+08] and statistical in [HMZ08]. Recall that the our dynamic adversary model is essentially a restriction of the general mixed adversary model to the threshold case. However, as also explained earlier, none of our positive results, nor negative results related to round complexity and notions weaker than G.O.D, are implied from [BFH+08,HMZ08].

1.5 Overview of the Document

In Section 2 we introduce some preliminaries, including notation and the dynamic security model we consider in this work. Then we proceed to presenting our main contributions. Sections 3 and 4 present our impossibility results for SFE/reactive MPC for statistical and perfect security, respectively. Then, we present feasibility results for statistical SFE, statistical MPC and perfect VSS in Section 5.1, 5.2 and 5.3 respectively. In Section 6 we give a detailed study of the general adversary results from [BFH+08] and [HMZ08] and what they imply for our case.

2 Preliminaries

2.1 Notation

In this work we consider a set $\mathcal{P} = \{P_1, \ldots P_n\}$ of n parties connected via synchronous and secure point-to-point channels. For the statistical setting, we additionally assume the presence of a broadcast channel. Let $\mathcal{A}^{\mathsf{stat}}$ and $\mathcal{A}^{\mathsf{perf}}$ denote a dynamic adversary who respects $2t_a+2t_p+t_f < n$ and $3t_a+2t_p+t_f < n$ respectively. Composition of two functions, f and g (say, $h(x) = g(f(x))$) is denoted as $g \circ f$. We use $[a, b]$ to denote the set $\{a, a+1, \ldots, b\}$, for $a \leq b$. We let $\mathbb{F}$ denote a field.

2.2 Security Model

In this work we consider the stand-alone security model [Can00]. A party can be either honest, passively corrupt, actively corrupt or fail-stop corrupt. Passively corrupt parties share their internal state with the adversary, but behave honestly.

The behavior of actively corrupt parties on the other hand is completely controlled by the adversary.

Fail-stop parties are modeled as a property of the underlying network: The adversary is allowed to specify, in every communication round, a subset of parties that are intended to fail-stop, meaning that they stop participating in the protocol. When a party is set to fail-stop by the adversary, it does not send *any* message to any honest party, which in turns enables honest parties to agree on which parties fail-crashed in a given round, as discussed below in Section 2.2.1. On the other hand, the adversary is not allowed to read the internal state of the fail-stop parties. However, he is allowed rushing fails i.e. the adversary can see the messages that fail-stop parties would have sent to corrupt parties in the round they are set to fail. This includes the messages sent over the broadcast channel in the statistical setting (which assumes the presence of an additional broadcast channel), even if no party is actively or passively corrupted. Notice that this does not happen in the perfect security setting since in this case the broadcast channel can be instantiated by protocols such as the efficient broadcast protocol of [AFM99] that is secure against dynamic adversaries. These protocols are executed directly on top of the secure point-to-point channels, so an adversary only corrupting fail-stop parties will not get access to any message in these channels.

A protocol is secure if a real-world execution as described above can be made indistinguishable by an ideal adversary (a.k.a. simulator) in an ideal execution. In such execution there is a trusted party who evaluates the intended function f faithfully. More precisely, all the parties begin by sending their input to a trusted party, and the adversary sends a subset of fail-stop parties $\mathcal{F}_I$. Then the trusted party evaluates f on these inputs, except it sets a default input for the parties in $\mathcal{F}_I$. This models the fact that the adversary may fail-stop some parties before they are even able to provide input. Next, the trusted party receives from the ideal adversary another subset of fail-stop parties $\mathcal{F}_O$. In the setting of fairness and abort security, the trusted party also receives from the ideal adversary a potential abort signal. In case of abort security, the trusted party would return the output of f to the adversary and relay this abort signal to all honest parties and abort. In case of fairness, only the latter occurs (i.e. abort signal is relayed but the output is not returned to the adversary). For reactive functionalities, the adversary can choose to activate the abort signal or not in each phase of the reactive functionality. In the setting of guaranteed output delivery (G.O.D.) such signal is not allowed. Finally, if the trusted party did not stop from an abort signal, it sends the output of f to the adversary and to the honest parties not in $\mathcal{F}_O$.

Let us denote the output of all the parties in the ideal and real executions by $\mathsf{IDEAL}_{f,\mathsf{S}}((x_i)_{i=1}^n)$ and $\mathsf{REAL}_{f,\mathsf{A}}((x_i)_{i=1}^n)$, where S and A are the ideal and real-world adversaries, and the x_i's are the inputs. A protocol securely evaluates the function f (with abort or fairness or G.O.D.) with perfect security if for every non-uniform probabilistic polynomial-time adversary A for the real model, there exists a non-uniform probabilistic polynomial-time adversary S for the ideal model, such that the distributions $\mathsf{IDEAL}_{f,\mathsf{S}}((x_i)_{i=1}^n)$ and $\mathsf{REAL}_{f,\mathsf{A}}((x_i)_{i=1}^n)$ are

identical for any set of inputs. The security is statistical, instead of perfect, if the statistical distance between these two distributions is negligible (in some statistical security parameter).

2.2.1 Detecting Fail-Stop Corruptions. If some party P_i does not receive a message by some other party P_j in a given round, then P_i cannot conclude that P_j is fail-corrupt since this behavior can be exhibited as well by actively corrupt parties (which may, for example, stop sending messages to only some subset of the parties). However, there is a simple method by which the parties can detect which parties fail-stop in a given round. After every round, an extra "heartbeat" round is added in which the parties must broadcast a constant bit which signals they are still "alive". If some party fails to broadcast such value, then it is considered as fail-stop.[3] Therefore, we assume that when an adversary fail-corrupts a party in a particular round, then his identity is exposed to all henceforth.

2.3 Definitions.

Verifiable Secret Sharing (VSS) [CGMA85]. A pair of protocols $(\pi_{\mathsf{Sh}}, \pi_{\mathsf{Rec}})$ for $\mathcal{P}$, where a dealer $D = P_1$ holds a private input $s \in \mathbb{F}$ (referred to as the secret) is a VSS scheme tolerating $\mathcal{A}$ if the following requirements hold for every possible $\mathcal{A}$ and for all possible inputs of D:

- *Correctness:* If D is honest, then the honest parties output s at the end of π_{Rec}. Moreover, this is true for any choice of the random inputs of the honest parties and $\mathcal{A}$'s randomness.
- *Strong Commitment:* If D is corrupted, then at the end of the sharing phase there is a value $s^* \in \mathbb{F}$ such that at the end of π_{Rec}, all honest parties output s^*, irrespective of the behavior of the corrupted parties.
- *Privacy:* If D is honest then $\mathcal{A}$'s view during π_{Sh} reveals no information on s. More formally, $\mathcal{A}$'s view is identically distributed for all different values of s

While in the perfect setting, no error is allowed, statistical VSS allows a negligible error in the properties of correctness and strong commitment.

3 Impossibility Results for Statistical Security

In this section, we present two negative results with respect to $\mathcal{A}^{\mathsf{stat}}$ i.e. a dynamic adversary who respects $2t_a + 2t_p + t_f < n$. First, we present a lower bound on the round complexity of statistical SFE (Section 3.1). Next, we present the impossibility for statistical VSS (more generally, reactive MPC) (Section 3.2).

[3] Observe that there may be false-positives, that is, parties who did not fail to send a message in the actual round, but failed to send the signal bit in the heartbeat round. However, this is acceptable in the protocols we consider in this work.

3.1 Secure Function Evaluation

We show that the price of non-constant round complexity ($\Omega(n)$ rounds) is necessary to design a statistical fair SFE against $\mathcal{A}^{\mathsf{stat}}$. We state the formal theorem below.

Theorem 1. *There exist standard (non-reactive) functionalities f such that any n-party (where $n \geq 4$) fair SFE protocol computing f with statistical security against a dynamic adversary must have $\Omega(n)$ rounds (specifically, at least $\frac{n}{4}+1$ rounds).*

Proof. We assume $n = 4\ell$ for simplicity, where $\ell \geq 1$. For the sake of contradiction, assume the existence of an r-round statistically-secure MPC protocol π computing a common output function f (that gives the same output to all) that achieves fairness against $\mathcal{A}^{\mathsf{stat}}$, where $r = \frac{n}{4}$.

Consider an execution of π on the set of inputs $(x_1, \ldots, x_n)$ and the following sequence of hybrids $\{\mathsf{H}_1, \ldots, \mathsf{H}_r\}$ described below. Each hybrid involves only active corruptions and rushing fails. In hybrid H_i, let $\mathcal{S}_a^i$, $\mathcal{S}_f^i$, $\mathcal{W}^i = \mathcal{P} \setminus (\mathcal{S}_a^i \cup \mathcal{S}_f^i)$ denote the set of active corruptions, fail-stop corruptions and honest parties respectively.

H_1: $\mathcal{A}^{\mathsf{stat}}$ chooses to corrupt a set $\mathcal{S}_a^1$ of $\frac{n}{4}$ parties actively, fail-stop corrupts a different set $\mathcal{S}_f^1$ of $(\frac{n}{2}-1)$ parties and then does the following: Behave honestly up to (and including) Round $r-1$. In Round r, fail-corrupt $\mathcal{S}_f^1$ and stay silent on behalf of $\mathcal{S}_a^1$.

H_2: $\mathcal{A}^{\mathsf{stat}}$ chooses to corrupt a set $\mathcal{S}_a^2 (= \mathcal{W}^1)$ of $(\frac{n}{4}+1)$ parties actively, fail-stop corrupts a different set of $\mathcal{S}_f^2$ of $(\frac{n}{2}-3)$ parties and does the following: Behave honestly (up to and including) Round $r-2$. In Round $r-1$, fail-corrupt $\mathcal{S}_f^2$ and stay silent on behalf of $\mathcal{S}_a^2$.

We generalize the above description to define the remaining sequence $\mathsf{H}_3, \ldots, \mathsf{H}_r$.

H_i: $\mathcal{A}^{\mathsf{stat}}$ chooses to corrupt a set $\mathcal{S}_a^i (= \mathcal{W}^{i-1})$ of $\frac{n}{4}+(i-1)$ parties actively, fail-stop corrupts a different set of $\mathcal{S}_f^i$ of $\frac{n}{2}-(2i-1)$ parties and does the following: Behave honestly (up to and including) Round $r-i$. In Round $r-i+1$, fail-corrupt $\mathcal{S}_f^i$ and stay silent on behalf of $\mathcal{S}_a^i$.

We present a sequence of lemmas to complete the proof. Let $\mu = \mathtt{negl}(\kappa)$ denote the negligible probability with which security of π fails (where κ denotes the statistical security parameter). Below, $(x_1, \ldots, x_n)$ denotes a specific combination of inputs that are fixed across all hybrids.

Lemma 1. *In H_1, $\mathcal{A}^{\mathsf{stat}}$ obtains $y = f(x_1, \ldots, x_n)$ with probability at least $1-\mu$.*

Proof. Since the dynamic adversary $\mathcal{A}^{\mathsf{stat}}$ started misbehaving only in the last round, he must have received the entire communication throughout the protocol (as per an execution where everyone is honest). Note that this includes the messages that the fail-corrupt parties send to the actively corrupt parties in the

last round as well (as we assume rushing fails). It now follows from correctness of π (which holds with overwhelming probability $1-\mu$) that $\mathcal{A}^{\mathsf{stat}}$ gets the output $y = f(x_1, \ldots, x_n)$ with probability at least $1-\mu$. Note that the output must be computed on the fixed set of inputs $(x_1, \ldots, x_n)$ as the view of $\mathcal{A}^{\mathsf{stat}}$ is identically distributed to an execution where everyone behaves honestly with respect to this set of fixed inputs. □

Lemma 2. *Suppose $\mathcal{A}^{\mathsf{stat}}$ obtains $y = f(x_1, \ldots, x_n)$ with probability at least $1-(i-1)\mu$ in H_{i-1} $(i \in \{2, \ldots, r\})$. Then, $\mathcal{A}^{\mathsf{stat}}$ in H_i can compute $y = f(x_1, \ldots, x_n)$ at the end of Round $(r-i+1)$ with probability at least $1-(i \times \mu)$.*

Proof. Consider H_{i-1}. Fairness dictates that when $\mathcal{A}^{\mathsf{stat}}$ obtains the output $y = f(x_1, \ldots, x_n)$ in H_{i-1} (assumed to occur with probability $1-(i-1)\mu$) [4], the honest parties should also be able to compute the same output $y = f(x_1, \ldots, x_n)$, even though parties in $(\mathcal{S}_a^{i-1} \cup \mathcal{S}_f^{i-1})$ stopped communicating after Round $(r-i+1)$. The honest parties constituting $\mathcal{W}^{i-1} = \mathcal{P} \setminus (\mathcal{S}_a^{i-1} \cup \mathcal{S}_f^{i-1})$ only interact amongst themselves after Round $(r-i+1)$. Since fairness breaks with probability at most μ, we can conclude that the combined view of parties in $\mathcal{W}^{i-1}$ at the end of Round $(r-i+1)$ must suffice to compute the output with probability at least $1-[(i-1)\mu+\mu] = 1-(i \times \mu)$.

Next, recall that $\mathcal{A}^{\mathsf{stat}}$ actively corrupts $\mathcal{S}_a^i = \mathcal{W}^{i-1}$ in H_i. We claim that the view of $\mathcal{A}^{\mathsf{stat}}$ in H_i is identically distributed to the combined view of parties in $\mathcal{W}^{i-1}$ in H_{i-1}. This is because $\mathcal{A}^{\mathsf{stat}}$ in H_i starts misbehaving only during Round $(r-i+1)$ and therefore must have received all incoming messages until Round $(r-i+1)$ as per an execution where everyone is honest. We can thus conclude that $\mathcal{A}^{\mathsf{stat}}$ in H_i can compute y at the end of Round $(r-i+1)$ with probability at least $1-(i \times \mu)$.

□

Lemma 3. *In H_i $(i \in \{1, \ldots, r\})$, $\mathcal{A}^{\mathsf{stat}}$ obtains $y = f(x_1, \ldots, x_n)$ at the end of Round $(r-i+1)$ with probability at least $1-(i \times \mu)$.*

Proof. The proof follows directly from Lemma 1 - 2. □

Lemma 4. *There exists an adversarial strategy that breaches security of π with overwhelming probability.*

Proof. It follows from Lemma 3 that $\mathcal{A}^{\mathsf{stat}}$ in H_r obtains $y = f(x_1, \ldots, x_n)$ at the end of Round 1 with probability at least $1-(r \times \mu) = 1-(\frac{n}{4} \times \mathtt{negl}(\kappa))$ which is overwhelming.

Thus, since $\mathcal{A}^{\mathsf{stat}}$ in H_r obtains output at the end of Round 1 itself, he can breach privacy of honest parties by executing the residual attack - Specifically, $\mathcal{A}^{\mathsf{stat}}$ can get multiple evaluations of f on various choices of inputs of corrupt parties, while the inputs of the honest parties remains fixed. This may allow

[4] Here, it is implicitly assumed that the function output depends on honest parties' inputs i.e. it could not have been computed locally by $\mathcal{A}^{\mathsf{stat}}$ using corrupt parties' inputs. Thereby, the argument for fairness can be invoked.

$\mathcal{A}^{\mathsf{stat}}$ to learn more information about the honest parties' inputs, beyond what is allowed in the ideal world (where the adversary gets the output only for a unique combination of inputs).

As a concrete example, suppose $f(x_1, \ldots, x_n)$ with $x_1 = (m_0, m_1)$, $x_i = b_i$ for $i = 2$ to n is defined as :

$$f(x_1, \ldots, x_n) = \begin{cases} m_0 & \text{if } \oplus_{i=2}^{n} b_i = 0 \\ m_1 & \text{otherwise} \end{cases}$$

where (m_0, m_1) denote a pair of messages and $b_i \in \{0, 1\}$ for $i \in \{2, \ldots, n\}$. Suppose P_1 is an honest party in H_r. Firstly, we point that f satisfies the implicit assumption mentioned earlier in Lemma 2 i.e. $\mathcal{A}^{\mathsf{stat}}$ (who does not corrupt P_1) cannot obtain the output of f using corrupt parties' inputs. Thus, the sequence of arguments above hold and there exists an adversarial strategy that allows $\mathcal{A}^{\mathsf{stat}}$ in H_r to obtain both m_0 and m_1 – the adversary can learn this by locally computing the output based on different choices of corrupt P_i's input i.e. $b_i = 0$ and $b_i = 1$. This attack breaches privacy of honest P_1. We have thus arrived at a contradiction; completing the proof of Theorem 1. □

Thus, $\Omega(n)$ rounds are necessary for fair statistically-secure MPC against a dynamic adversary.

3.2 Reactive MPC

We present the feasibility of achieving reactive MPC with G.O.D against $\mathcal{A}^{\mathsf{stat}}$ below, which also follows from the results in [HMZ08].

Theorem 2. *Let $\mathcal{S}$ denote the set of corruption strategies that the dynamic adversary can choose from. In the statistical setting, reactive MPC (such as VSS) with G.O.D is impossible against a dynamic adversary if $R_{\mathcal{S}} + F_{\mathcal{S}} \geq n$, where $R_{\mathcal{S}}$ is the maximal number of player states the adversary can read, while $F_{\mathcal{S}}$ is the maximal number of players the adversary can have abort the protocol.*

Proof. Assume by contradiction that there exists a statistical VSS $\pi = (\pi_{\mathsf{Sh}}, \pi_{\mathsf{Rec}})$ (where π_{Sh} and π_{Rec} denote sharing and reconstruction protocols respectively) that achieves G.O.D against a dynamic adversary $\mathcal{A}^{\mathsf{stat}}$ who can choose any strategy from $\mathcal{S}$, where $R_{\mathcal{S}} + F_{\mathcal{S}} \geq n$. Suppose $\mathcal{A}^{\mathsf{stat}}$ behaves honestly during π_{Sh} which completes successfully and then fail-crashes $F_{\mathcal{S}} \geq n - R_{\mathcal{S}}$ parties during π_{Rec}. This would violate G.O.D as the secret cannot be determined from the state of the remaining $n - F_{\mathcal{S}} \leq R_{\mathcal{S}}$ parties (otherwise, the adversary could have learnt the secret as it can read the state of up to $R_{\mathcal{S}}$ parties).

□

The above result shows that $\mathcal{A}^{\mathsf{stat}}$ must satisfy the additional condition of $R_{\mathcal{S}} + F_{\mathcal{S}} < n$ for VSS with G.O.D to be feasible. In fact, this condition is not only necessary, but also sufficient for dynamic VSS and reactive MPC with G.O.D as shown by our construction in Appendix C.1

Lastly, we remark that the above argument can be viewed in terms of (t_a, t_p, t_f) as the condition $2t_a + 2t_f \leq n$ being necessary for VSS with G.O.D (in addition to $2t_a + 2t_p + t_f < n$ respected by $\mathcal{A}^{\mathsf{stat}}$) against $\mathcal{A}^{\mathsf{stat}}$. This is because if $2t_a + 2t_f > n$, an adversary aborting on behalf of $t_a + t_f > n/2$ parties during π_{Rec} violates G.O.D; as the secret cannot be determined from the state of the remaining $n - t_a - t_f < n/2$ parties (follows from privacy during π_{Sh}).

4 Impossibility Results for Perfect Security

In this section, we present two negative results with respect to $\mathcal{A}^{\mathsf{perf}}$ i.e a dynamic adversary who respects $3t_a + 2t_p + t_f < n$ - We prove the impossibility of perfect SFE with abort and perfect VSS with G.O.D against $\mathcal{A}^{\mathsf{perf}}$ in Section 4.1 and 4.2 respectively.

4.1 Secure Function Evaluation

We show that perfect dynamic SFE is impossible. In fact, our impossibility argument is stronger than the above statement in two aspects - First, it holds even if the perfect SFE protocol against $\mathcal{A}^{\mathsf{perf}}$ is only required to achieve the weaker security notion of security with abort (adversary may get the output while honest parties do not; implied by fairness and G.O.D). Second, it holds even against a weaker dynamic adversary who is allowed only active and passive corruptions (i.e. $t_f = 0$).

Theorem 3. *There exists a standard (non-reactive) functionality f for which no n-party protocol computing f can achieve perfect security with abort (implied by fairness and G.O.D) against a dynamic adversary, even if $t_f = 0$.*

Proof. We present the argument for $n = 5$ for simplicity. The proof can be extended in a natural manner for $n > 6$ (elaborated in Appendix A.2).

For the sake of contradiction, we assume a protocol π that achieves perfect security with abort against $\mathcal{A}^{\mathsf{perf}}$ and computes the function $f(x_1, x_2, x_3, x_4, x_5)$ among the set of parties $\{P_1, P_2, P_3, P_4, P_5\}$. Here x_i denotes P_i's input where x_1 and x_2 are single bit values and $x_3 = x_4 = x_5 = \perp$. Suppose f computes $(x_1 \wedge x_2)$ i.e the logical AND of the input bits of P_1 and P_2.

Next, we present the transformation of the 5-party perfectly secure protocol π computing f to a 3-party perfectly secure protocol π' that computes $f'(x'_1, x'_2, x'_3)$ among $\{P_1^*, P_2^*, P_3^*\}$. Here x'_i denotes the input of P_i^* where x'_1 and x'_2 are single bit values and $x'_3 = \perp$. Let f' be defined as the logical AND of the input bits of P_1^* and P_2^* i.e $(x'_1 \wedge x'_2)$. π' proceeds as follows:

- P_1^* emulates the role of $\{P_1, P_3\}$ in π using input $x'_1 = x_1$.
- P_2^* emulates the role of $\{P_2, P_4\}$ in π using input $x'_2 = x_2$.
- P_3^* emulates the role of P_5 in π using input $\perp$.

It follows from correctness of π that π' should result in correct output $(x'_1 \wedge x'_2)$ and thereby computes f'. Next, recall that π can tolerate up to 2 passive corruptions or 1 active corruption among 5 parties (satisfying $3t_a + 2t_p < 5$) and therefore must be secure in scenarios of **(a)** passive corruptions of $\{P_1, P_3\}$ **(b)** passive corruptions of $\{P_2, P_4\}$ and **(c)** active corruption of P_5. It is easy to check from the transformation that these scenarios translate to **(a)** passive corruption of P_1^* **(b)** passive corruption of P_2^* and **(c)** active corruption of P_3^* respectively. We can thus conclude that π' achieves security with abort against an adversary who can choose among the above 3 corruption options. However, this contradicts the impossibility result of [FHM98] (elaborated in Appendix A.1) which proves that no 3-party perfectly-secure protocol (achieving security with abort) among $\{P_1^*, P_2^*, P_3^*\}$ computing $(x'_1 \wedge x'_2)$ can be secure against an adversary that passively corrupts either P_1^* or P_2^* or actively corrupts P_3^*. We have thus arrived at a contradiction, completing the proof of Theorem 3. □

Lastly, we point that the above argument exploits only active and passive corruptions, and thereby holds even when $t_f = 0$. This is in contrast to the scenarios of other weaker dynamic adversaries with $t_a = 0$ and $t_p = 0$ as demonstrated by our upper bounds in Appendix C.2.

4.2 Reactive MPC

Here, we observe that the non-dynamic feasibility condition $3t_a + 2t_p + t_f < n$ is not sufficient for perfect VSS, not even if $t_f = 0$.

Theorem 4. *The requirement $3t_a + 2t_p < n$ does not allow for perfect VSS with G.O.D against a dynamic adversary, when $n \geq 7$.*

This follows from Lemma 14 in Section 6. It shows that even if we assume $3t_a + 2t_p < n$ it can still be the case that the C_{rec} condition from [BFH+08] is violated, and this condition was shown in [BFH+08] to be required for robust reconstruction of a secret shared value.

The feasibility of dynamic perfectly-secure VSS with G.O.D for the special cases of $t_a = 0$ and $t_p = 0$ are investigated in Appendix C.3.

5 Positive Results

5.1 SFE with Statistical Security

Let f be an n-input function with a single output. In this section we present a statistically secure protocol against a dynamic adversary that has G.O.D. and uses at most $O(n)$ rounds, regardless of the complexity of the function f. We begin by introducing in Sections 5.1.1 and 5.1.2 the necessary building blocks for our protocol from Section 5.1.3, namely robust sharings and levelled sharings, respectively. The former sharings are useful for secret-sharing a value while ensuring that, at reconstruction time, either the honest parties get the secret

Function $f_{\mathsf{sh}}^{d,\mathsf{stat}}(s)$

Implicit input: $\mathcal{Q} \subseteq \mathcal{P}$.

1. Sample a random polynomial $g(x) \in \mathbb{F}[x]$ of degree at most d such that $g(0) = s$, where $\mathbb{F}$ denotes a finite field with $|\mathbb{F}| > n$.
2. Let $s_i = f(i)$ for $P_i \in \mathcal{Q}$.
3. For i, j such that $P_i, P_j \in \mathcal{Q}$, sample $K_{ji} = (\alpha_{ji}, \beta_{ji}) \in \mathbb{F}^2$ and let $m_{ij} = \alpha_{ji} \cdot s_i + \beta_{ji}$.
4. Let b_i be the tuple $(s_i, \{m_{ij}\}_{j=1}^n, \{K_{ij}\}_{j=1}^n)$.
5. Output $(b_i)_{P_i \in \mathcal{Q}}$, where b_i is intended for party P_i.

Fig. 2. Functionality for generating Shamir sharings together with authentication information

or they output a set of identified corrupt parties, whereas the latter sharings are used to ensure that this reconstruction is done in a fair way, that is, if the adversary disallows the honest parties from learning the secret (which identifies some corrupt parties in the process), the adversary cannot get the secret himself.

5.1.1 Robust Sharings. At the core of our techniques lies the ability of the honest parties to identify which shares are correct when opening some secret-shared value. This is captured by the function $f_{\mathsf{sh}}^{d,\mathsf{stat}}(s)$, presented in Fig. 2, which produces the shares of a secret s together with the additional information that the parties need to identify incorrect shares. This technique is motivated by the VSS in [RB89]. The function takes an implicit parameter $\mathcal{Q} \subseteq \mathcal{P}$ that, as we will see later on, denotes the actual set of parties among which the computation takes place.

Throughout the rest of this section we denote by $[s]_d$ the output of $f_{\mathsf{sh}}^{d,\mathsf{stat}}(s)$ produced by an ideal functionality, where the set $\mathcal{Q}$ is implicit from context. The protocol π_{StatRec}^d in Fig. 3 is used by the parties to reconstruct a shared value $[s]_d$. The protocol guarantees that the parties either reconstruct the secret correctly, or they output a set of corrupt parties who misbehaved in the protocol. The protocol also takes as an additional input a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ among which the secret is shared and who will participate in the protocol. We denote by t'_a, t'_p and t'_f the number of active, passive and fail-stop corrupt parties in $\mathcal{Q}$, and we write $n' = |\mathcal{Q}|$. As we will see later, the idea is that the parties in $\mathcal{P} \setminus \mathcal{Q}$ are parties who have been previously identified as corrupt, so they will not participate in the current reconstruction. In particular, the bound $2t'_a + 2t'_p + t'_f < n'$ also holds for the set $\mathcal{Q}$.

Before we prove the security properties of π_{StatRec}^d, we present the following useful lemma. Its proof is standard and is presented in Section B in the appendix.

Protocol $\pi^{d}_{\mathsf{StatRec}}$

Input: A shared value $[s]_d$ among a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ where $2t'_a + 2t'_p + t'_f < n'$.
Output: Secret s or $\perp$ with two sets $\mathcal{A}, \mathcal{F} \subseteq \mathcal{Q}$ of identified active and fail-stop corrupt parties, respectively.

1. Each party $P_i \in \mathcal{Q}$ broadcasts its share s_i together with $\{m_{ij}\}_{j=1}^{n}$.
2. Let $\mathcal{F}_1$ be the set of parties who fail-stopped during the first round above. If $|\mathcal{Q} \setminus \mathcal{F}_1| \leq d$, then output $\perp$ together with the pair of sets $(\emptyset, \mathcal{F}_1)$.
3. Else, each party $P_j \in \mathcal{Q} \setminus \mathcal{F}_1$, having $\{K_{ji} = (\alpha_{ji}, \beta_{ji})\}_{i=1}^{n}$, checks for i such that $P_i \in \mathcal{Q} \setminus \mathcal{F}_1$ whether $m_{ij} \stackrel{?}{=} \alpha_{ji} \cdot s_i + \beta_{ji}$ holds. For every i that does not satisfy this equality, P_j broadcasts (accuse, P_i).
4. Let $\mathcal{F}_2 \subseteq \mathcal{Q} \setminus \mathcal{F}_1$ be the set of parties who fail-stopped during the previous "accusation" round. Initially all parties set $\mathcal{A} = \emptyset$. For every party P_i such that at least $\lceil (n'' + 1)/2 \rceil$ messages (accuse, P_i) were broadcasted, where $n'' = n' - |\mathcal{F}_1| - |\mathcal{F}_2|$, all parties in $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$ add P_i to $\mathcal{A}$.
5. If $|\mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F}_1 \cup \mathcal{F}_2)| > d$, then use the shares $\{s_i\}_{P_i \in \mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F}_1 \cup \mathcal{F}_2)}$ to reconstruct s using polynomial interpolation. Else, output $\perp$ and the pair $(\mathcal{A}, \mathcal{F}_1 \cup \mathcal{F}_2)$.

Fig. 3. Protocol for reconstructing Shamir sharings with authentication information

Lemma 5. *Consider an actively corrupt party P_i and an honest party P_j in $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$ in protocol $\pi^{d}_{\mathsf{StatRec}}$. Let s_i be P_i's share in $[s]_d$, and suppose P_i broadcasts $s'_i \neq s_i$ in the first step. Then, with probability at least $1 - \frac{1}{|\mathbb{F}|}$, P_j broadcasts (accuse, P_i) in the accusation round.*

With the lemma at hand it is easy to prove the following proposition, which presents the properties of $\pi^{d}_{\mathsf{StatRec}}$.

Proposition 1. *Suppose a robust sharing $[s]_d \leftarrow f^{d,\mathsf{stat}}_{\mathsf{sh}}(s)$ is used as an input to $\pi^{d}_{\mathsf{StatRec}}$ and assume that $|\mathbb{F}| > 2^{\kappa}$.*[5] *If a value s' is produced as the output then, with overwhelming probability, it holds that $s' = s$. Otherwise, if $\perp$ is the output, then the sets $\mathcal{A}$ and $\mathcal{F}$ produced by the protocol consist of exactly the malicious parties who lied about their share or MAC and the fail-stop parties, respectively. In particular, $|\mathcal{Q}| - |\mathcal{A} \cup \mathcal{F}| \leq d$.*

Proof. Let $[s]_d = \big((s_i, \{m_{ij}\}_{j=1}^{n}, \{K_{ij}\}_{j=1}^{n})\big)_{i=1}^{n}$. We begin by proving that the set $\mathcal{A}$ computed by the parties after the accusation phase consists of exactly the parties who lied about their share, with overwhelming probability. To see that every party who lies about his share is included in this set consider a malicious party P_i who engages in such behavior. Due to Lemma 5, all honest

[5] This restriction is easily removed by modifying the sharing mechanism to include multiple key-tag pairs.

Function $f^{\alpha,\beta}_{\mathsf{sLevSh}}(s)$

Implicit input: $\mathcal{Q} \subseteq \mathcal{P}$.
Output: (α,β)-levelled-sharing of s denoted by $\langle s \rangle^{\alpha,\beta}$.

1. Sample a random elements $s_\alpha, \ldots, s_\beta \in \mathbb{F}$ such that $\sum_{d=\beta}^{\alpha} s_d = s$
2. For $d = \beta, \ldots, \alpha$ call $[s_d]_d = f^{d,\mathsf{stat}}_{\mathsf{sh}}(s_d)$ using the set $\mathcal{Q}$.
3. Output $([s_\beta]_\beta, \ldots, [s_\alpha]_\alpha)$, where the i-th entry in each $[s_d]_d$ is intended for party P_i.

Fig. 4. Functionality to generate levelled-sharings of a secret

parties in $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$ broadcast (accuse, P_i) with overwhelming probability. Furthermore, we know that $t'_f \geq |\mathcal{F}_1| + |\mathcal{F}_2|$ and also $2t'_a + 2t'_p + t'_f < n'$, so $2t'_a + 2t'_p < n' - |\mathcal{F}_1| - |\mathcal{F}_2| = n''$. In particular, there are at least $\lceil \frac{n''+1}{2} \rceil$ honest parties in $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$, so P_i will get enough accusations to be put in $\mathcal{A}$.

In the opposite direction, we now argue that no honest party is placed in $\mathcal{A}$. For this, it suffices to observe that no honest party will accuse another honest party, and the adversary can produce at most $\lfloor (n''-1)/2 \rfloor$ accusations, which is strictly less than the minimal number of accusations required for placing a party in $\mathcal{A}$.

With the above analysis at hand it is easy to prove the proposition: The shares of parties from $\mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F}_1 \cup \mathcal{F}_2)$ are correct, so if there are at least $d+1$ of them the secret can be reconstructed correctly. If reconstruction is not possible it is because there are not enough shares, that is, $|\mathcal{Q}| - |\mathcal{A} \cup \mathcal{F}_1 \cup \mathcal{F}_2| \leq d$. □

5.1.2 Levelled Sharings. Proposition 1 shows that an adversary cannot make the honest parties reconstruct an incorrect value without revealing the identity of some of the corrupt parties. However, a negative aspect of the protocol π^d_{StatRec} above is that it is not fair: The adversary can learn the secret after the parties broadcast their shares, and it can send incorrect shares so that the other parties do not learn the secret. To obtain a fair reconstruction protocol we use the levelled-sharing idea from [HLM13,PR19], by which a secret is shared first additively, and then each additive share is distributed using the sharing function $f^{d,\mathsf{stat}}_{\mathsf{sh}}$ from above, parameterized by different degrees.

We present the details of this technique below. First we define the function $f^{\alpha,\beta}_{\mathsf{sLevSh}}(s)$ that is analogous to $f^{d,\mathsf{stat}}_{\mathsf{sh}}(s)$ and takes care of generating levelled shares of the secret s. This function, presented in Fig. 4, is parameterized by two positive integers $\alpha \geq \beta$, and produces $[\cdot]$-sharings of additive shares of secret using degrees that vary from β to α. As $f^{d,\mathsf{stat}}_{\mathsf{sh}}$, $f^{\alpha,\beta}_{\mathsf{sLevSh}}$ also accepts a set $\mathcal{Q} \subseteq \mathcal{P}$.

Similarly to $[s]_d$ and $f^{d,\mathsf{stat}}_{\mathsf{sh}}$, we denote by $\langle s \rangle^{\alpha,\beta}$ the output of $f^{\alpha,\beta}_{\mathsf{sLevSh}}(s)$ produced by an ideal functionality. Notice that these sharings preserve the privacy of the secret as long as the adversary controls at most α shares, since this

Protocol $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$

Input: A shared value $\langle s\rangle^{\alpha,\beta}$ and a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ where $2t'_a+2t'_p+t'_f < n'$.
Output: Secret s or $\perp$ with two sets $\mathcal{A}, \mathcal{F} \subseteq \mathcal{Q}$ of identified active and fail-stop corrupt parties, respectively.

1. For $d = \alpha, \ldots, \beta$ each party $P_i \in \mathcal{Q}$ does the following.
   (a) Call $\pi^{d}_{\mathsf{StatRec}}([s_d]_d)$.
   (b) If the output is s_d, then continue. Else, if the output is $\perp$ and the pair of sets $(\mathcal{A}, \mathcal{F})$, then stop and output $\perp$ together with the pair $(\mathcal{A}, \mathcal{F})$.
2. Output $s = s_\alpha + \cdots + s_\beta$.

Fig. 5. Protocol for reconstructing levelled-sharings

implies that the adversary cannot learn the additive share s_α. Protocol $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$ in Fig. 5, which is analogous to $\pi^{d}_{\mathsf{StatRec}}$, shows how the parties can reconstruct at $\langle\cdot\rangle^{\alpha,\beta}$-sharing while satisfying fairness, that is, either all parties learn the secret correctly or no one does. This, together with other properties of $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$, is formalized in Proposition 2 below.

Proposition 2. *Assume that an (α, β)-levelled sharing $\langle s\rangle^{\alpha,\beta} \leftarrow f^{\alpha,\beta}_{\mathsf{sLevSh}}(s)$ is used as input in protocol $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$. Then the following holds:*

- ***Correctness.*** *If the parties output a value different to $\perp$, then this value equals the correct secret s.*
- ***Fault-Identification.*** *If the adversary disrupts the reconstruction of $[s_d]_d$, then the parties output a pair of sets $\mathcal{A}, \mathcal{F} \subseteq \mathcal{Q}$ of actively and fail-stop corrupt parties, respectively, where $|\mathcal{F} \cup \mathcal{A}| \geq |\mathcal{Q}| - d$.*
- ***Fairness.*** *If the opening of $[s_j]$ (where $j > \beta$) results in abort, then the adversary does not learn s_{j-1}.*

Proof. Correctness and fault-identification follow directly from Proposition 1, so it suffices to show the fairness property. First, we assume, for simplicity in the notation, that $t_p = 0$. This is without loss of generality since active and passive corruptions cost the same to the adversary, but passive corruptions are less powerful. We begin by noticing that, in protocol $\pi^{d}_{\mathsf{StatRec}}$, the honest parties fail to open $[s_j]_j$ if the set $\mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F}_1 \cup \mathcal{F}_2)$ has at most j parties, that is, if $n' - |\mathcal{A}| - |\mathcal{F}_1| - |\mathcal{F}_2| \leq j$. On the other hand, for the adversary to learn s_{j-1} it needs to obtain at least j shares. We note that while the dynamic adversary (with rushing fails) who disrupts the reconstruction of s_j would be able to see the messages (i.e. the shares) of the fail-corrupt parties corresponding to s_j ; he would not be able to see their shares corresponding to s_{j-1}. Therefore, the adversary would have access to only t'_a shares (the ones corresponding to the actively corrupt parties in $\mathcal{Q}$) corresponding to s_{j-1} implying that $t'_a \geq j$ must hold for the adversary to learn s_{j-1}. However, since $2t'_a + t'_f < n' = \mathcal{Q}$, $|\mathcal{F}_1| + |\mathcal{F}_2| \leq t'_f$

Protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$

Inputs: Party P_i has input x_i, for $i = 1, \ldots, n$.
Output: $y = f(x_1, \ldots, x_n)$.
Building blocks: Function $f^{\alpha,\beta}_{\mathsf{sLevSh}}$ (Fig 4) and Protocol $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$ (Fig 5)

Initialize $\mathcal{Q} := \mathcal{P}$, $n' = n$.

1. If $n' \geq 3$, parties in $\mathcal{Q}$ use π_{StatBase} to compute $(r, \langle \widehat{y} \rangle^{\lceil n'/2 \rceil - 1, 1})$, where r denotes a random element in $\mathbb{F}$, $\widehat{y} = f(x_1, \ldots, x_n) + r$ (default inputs used for parties in $\mathcal{P} \setminus \mathcal{Q}$) and $\langle \widehat{y} \rangle^{\lceil n'/2 \rceil - 1, 1} \leftarrow f^{\lceil n'/2 \rceil - 1, 1}_{\mathsf{sLevSh}}(\widehat{y})$. Else, parties in $\mathcal{Q}$ use π_{StatBase} to compute $(r, \widehat{y})$ directly (as there are no active / passive corruptions, only potential fail-stop corruptions).
   - If at any round a party $P_i \in \mathcal{Q}$ is detected as fail-stop, then the parties update $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{P_i\}$, $n' = |\mathcal{Q}|$ and repeat step 1.
   - Else, the parties in $\mathcal{Q}$ obtain $(r, \langle \widehat{y} \rangle^{\lceil n'/2 \rceil - 1, 1})$.
2. Parties in $\mathcal{Q}$ run $\pi^{\lceil n'/2 \rceil - 1, 1}_{\mathsf{sLevRec}}(\langle \widehat{y} \rangle^{\lceil n'/2 \rceil - 1, 1})$. If it returns the value $\widehat{y}$ then the parties output $y = \widehat{y} - r$. Else, the protocol outputs a pair of sets $\mathcal{A}, \mathcal{F} \subseteq \mathcal{Q}$. Parties update $\mathcal{Q} \leftarrow \mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F})$, $n' = |\mathcal{Q}|$ and repeat step 1.

Fig. 6. Protocol for SFE with statistical security and G.O.D.

and $|\mathcal{A}| \leq t'_a$, we have that $t'_a < n' - t'_a - t'_f \leq n' - |\mathcal{A}| - |\mathcal{F}_1| - |\mathcal{F}_2| \leq j$, so we conclude that the adversary cannot reconstruct s_{j-1}. □

5.1.3 A Protocol with GOD. Now we are ready to describe our main protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$, which appears in Fig. 6 and is inspired in the protocol from [PR19], that achieves GOD with low round complexity by first executing a constant-round protocol with identifiable abort to compute levelled sharings and then performing a gradual opening of these levelled sharings. In the protocol we let π_{StatBase} be a constant-round *non-dynamic* statistically secure protocol with G.O.D. against a dishonest minority, which can be instantiated for example using randomizing polynomials [IK02], together with a non-constant round protocol like [BFO12], or the more efficient and recent protocol from [GSZ20]

While [PR19] involves levelled sharings of the output of f, we use the constant round protocol π_{StatBase} to choose a random element (to be used as a mask) and compute levelled sharings of the "masked" output. The mask is given on clear to the honest parties as output of π_{StatBase} (along with the levelled sharings) but would not be available to an adversary that performs *only* fail-stop corruptions. Looking ahead, this modification helps us tolerate rushing fails in the last round of the protocol.

Lemma 6. *Protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$ terminates in $O(n)$ rounds.*

Proof. To prove the lemma, we show via an inductive argument that the round complexity of $\pi^{\mathsf{stat}}_{\mathsf{god}}$ when executed among n' parties is bounded by $Rn' + n'$ [6] where R is a constant denoting the round complexity of π_{StatBase}.

Base Case: Suppose $n' = 1$ or 2. Then, it follows from the protocol description that the parties participate in π_{StatBase} to compute $(r, \widehat{y})$ directly, which may result in abort at most once (when 1 fail-corruption occurs corresponding to $n' = 2$). Thus, it is easy to see that $\pi^{\mathsf{stat}}_{\mathsf{god}}$ terminates in less than $Rn' + n'$ rounds; completing the base case.

Strong Induction Hypothesis ($n' \leq k$): Next, suppose that the statement is true for $n' \leq k$ parties.

Induction Step ($n' = k + 1$): Consider an execution of $\pi^{\mathsf{stat}}_{\mathsf{god}}$ among $n' = k + 1$ parties. Then, there are 3 exhaustive possibilities:

First, suppose neither Step 1 nor Step 2 fails. Note that Step 2 incurs round complexity $2(\lceil n'/2 \rceil - 1) < n'$ (as $\pi^{\lceil n'/2 \rceil - 1, 1}_{\mathsf{sLevRec}}(\cdot)$ involves $\lceil n'/2 \rceil - 1$ invocations of the 2-round subprotocol $\pi^{d}_{\mathsf{StatRec}}$). Thus, the total round complexity over Step 1 and Step 2 is bounded by $R + n' < n'R + n'$.

Next, suppose Step 1 fails. Then, it must be the case that at least one fail-corrupt party is eliminated and the protocol is re-run among $n' - 1 = k$ parties. Therefore, the round complexity is at most R (for the failed run) + $(kR + k)$ (via induction hypothesis) which totals up to $(k + 1)R + k < n'R + n'$.

Lastly, suppose Step 1 succeeds but Step 2 fails during the reconstruction of summand $\widehat{y}_i$ ($i \in [1, \lceil n'/2 \rceil - 1]$). From Proposition 2, it holds that at least $n' - i$ parties are eliminated and thereby at most i parties participate in the next re-run. Therefore, the round complexity is R (for Step 1) + $2(\lceil n'/2 \rceil - i)$ (for Step 2 of the failed run) + $iR + i$ (induction hypothesis for $i \leq k$ parties) which totals up to $(i + 1)R + 2\lceil n'/2 \rceil - i < n'R + n'$. This completes the induction step.

This completes the proof via induction that the statement is true for all $n' \geq 1$. We can thus conclude that $\pi^{\mathsf{stat}}_{\mathsf{god}}$, when executed among n parties, terminates within $Rn + n = O(n)$ rounds. □

Theorem 5. *Protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$ evaluates the function f in $O(n)$ rounds with statistical security against $\mathcal{A}^{\mathsf{stat}}$.*

The formal simulation-based proof of this theorem appears in Section B in the appendix. However, here we provide an intuition for the security argument. First, as we saw in Lemma 6, the protocol produces output within $O(n)$ rounds. However, in order to maintain privacy, it must be the case that before every re-run the adversary is not able to learn anything about the honest parties' inputs (else, $\mathcal{A}^{\mathsf{stat}}$ may be able to carry out a residual attack, for example, by using different inputs for the corrupt parties).

To see the adversary learns nothing right before a re-run, we argue informally as follows. First, if the re-run happens in the middle of the execution of π_{StatBase}, $\mathcal{A}^{\mathsf{stat}}$ does not learn anything because of the privacy of the protocol. Also, if the re-run takes place at the end of this protocol, then privacy is maintained because

[6] This is a loose bound chosen for simplicity as it suffices for our purpose.

of the privacy of the sharings $\langle \widehat{y} \rangle^{\lceil n'/2 \rceil - 1, 1}$, given that $\mathcal{A}^{\mathsf{stat}}$ gets to see at most $\lceil n'/2 \rceil - 1$ sharings at this stage.

Now we analyze what happens if the re-run takes place due to failure in the reconstruction of some $[\widehat{y_d}]_d$. If $d > 1$, then the privacy of the output is maintained since, from the fairness property in Proposition 2, disrupting the reconstruction of $[\widehat{y_d}]_d$ makes the adversary unable to learn the additive share $\widehat{y_{d-1}}$, which is necessary to learn $\widehat{y}$. Now, suppose reconstruction of $[\widehat{y_1}]_1$ is the first to fail, then the fault identification property in Proposition 2 and the condition $2t'_a + 2t'_p + t'_f < n'$ imply that $1 \geq |\mathcal{Q}| - |\mathcal{F} \cup \mathcal{A}| \geq n' - t'_f - t'_a > t'_a + 2t'_p$. This implies that $t'_a = t'_p = 0$ and $t'_f = n' - 1$ must hold. More specifically, $\mathcal{A}^{\mathsf{stat}}$ must have disrupted reconstruction of $[\widehat{y_1}]_1$ using $(n'-1)$ fail-stop corruptions. In this case, since $\mathcal{A}^{\mathsf{stat}}$ has access to the messages sent over the broadcast channel during the reconstruction of all summands, including the shares broadcast by the fail-corrupt parties during reconstruction of $[\widehat{y_1}]_1$, he would be able to learn $\widehat{y}$. However, we argue that fairness is still maintained as $\mathcal{A}^{\mathsf{stat}}$ (with $t'_a = t'_p = 0$ and $t'_f = n - 1$) does not have access to the internal state of any party (recall that the adversary is not allowed to read the internal state of the fail-stop parties). In particular, this means that even if the adversary participates honestly during π_{StatBase} (i.e. does not make any of the fail-stop parties crash), still he does not learn the output of π_{StatBase} and thereby the random mask r. This is because the output of π_{StatBase} cannot be learned from just the public transcript of the protocol but also requires the internal state of at least one participant. We can thus infer that $\mathcal{A}^{\mathsf{stat}}$ has no information about the random mask r, which is necessary to learn the output $y = \widehat{y} - r$. This completes the intuition.

Lastly, we analyze the complexity of the protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$. It is easy to see that if the subprotocol π_{StatBase} is instantiated using an efficient protocol, then $\pi^{\mathsf{stat}}_{\mathsf{god}}$ would have polynomial complexity (with complexity around n times that of π_{StatBase}). Since efficient non-constant round protocols [BFO12,GSZ20] exist for all functions, our construction is always efficient if we do not insist on asymptotically tight (but still polynomial) round complexity. This strictly improves over the constructions in [HMZ08] which have complexity exponential in n.

However, if π_{StatBase} is constant round, then we get $O(n)$ rounds which is asymptotically tight. Such a constant-round protocol exists for all functions but is not always computationally efficient. As mentioned in the introduction, it would be extremely surprising if tightness of $O(n)$ rounds could be shown with computational efficiency for all functions (as that would imply constant-round, information theoretically secure and computationally efficient protocol for all functions when n is a constant, which is a longstanding open question). On the other hand, if the function in question has an efficient non-dynamic constant-round protocol, as many functions do, then we can use that one to instantiate π_{StatBase} and get an efficient dynamic $O(n)$-round protocol.

5.2 Fair VSS with Statistical Security

We saw in Section 3.2 that dynamic VSS with G.O.D. and statistical security is impossible (without any additional restrictions). However, we observe that the ideas of Section 5.1.3 can be extended to design a fair VSS.

For the sharing protocol, the parties execute π_{StatBase} (a *non-dynamic* statistically secure protocol with G.O.D. against a dishonest minority) to compute $(r, \langle \widehat{s} \rangle^{\lceil n/2 \rceil - 1,1})$, where $\langle \widehat{s} \rangle^{\lceil n/2 \rceil - 1,1}$ represents the levelled-sharing of the "masked" secret $\widehat{s} = s + r$, with s and r denoting the dealer's input and the random mask respectively. For reconstruction, parties execute $\pi^{\lceil n/2 \rceil - 1,1}_{\mathsf{sLevRec}}(\langle \widehat{s} \rangle^{\lceil n/2 \rceil - 1,1})$. If any of the steps fail, the parties simply output $\perp$ (re-runs can be avoided as the goal is to achieve fairness). Else, the parties obtain $\widehat{s}$ and output the secret $s = \widehat{s} - r$. It is easy to check that privacy in case of honest dealer holds (as $\mathcal{A}^{\mathsf{stat}}$ controls at most $\lceil n/2 \rceil - 1$ parties actively / passively). Fairness and correctness of reconstruction follow directly from fairness and correctness of $\pi^{\lceil n/2 \rceil - 1,1}_{\mathsf{sLevRec}}(\cdot)$ (Proposition 2). Lastly, fairness is also maintained against an adversary who disrupts reconstruction of the last summand (i.e. the summand $[\widehat{s}_1]_1$) during $\pi^{\lceil n/2 \rceil - 1,1}_{\mathsf{sLevRec}}(\langle \widehat{s} \rangle^{\lceil n/2 \rceil - 1,1})$. Recall that this scenario occurs only when $t_a = t_p = 0$ and $t_f = n - 1$ (elaborated in the informal argument of Theorem 5). In such a case, the adversary learns $\widehat{s}$ but fairness is maintained as the adversary has no information about the random mask r (as the output of π_{StatBase} can be learnt only if adversary has access to internal state of at least one participant), and thereby the secret s.

The above result is summarized in the following theorem:

Theorem 6. *In the statistical setting, there exists a VSS with fair reconstruction against the dynamic adversary* $\mathcal{A}^{\mathsf{stat}}$.

Using the standard technique of verifiably secret-sharing the intermediate states [HMZ08], the above VSS and the SFE upper bound of Section 5.1.3 can be used to obtain a reactive MPC achieving fairness against $\mathcal{A}^{\mathsf{stat}}$.

Theorem 7. *In the statistical setting, there exists a fair MPC that can compute any reactive functionality against the dynamic adversary* $\mathcal{A}^{\mathsf{stat}}$.

5.3 Fair VSS with Perfect Security

In this section we present a VSS protocol with fair reconstruction against $\mathcal{A}^{\mathsf{perf}}$. The protocol design uses as a building block the modified variant of the VSS protocol of [BGW88] (modification proposed in [Dwo90,DDWY93]). While it is used for a fixed (t_a, t_p) in the work of [FHM98], we tweak the construction for security against dynamic adversary.

The biggest issue appears in making reconstruction fair. A similar situation was faced in Section 5.1.1 in the statistical setting, where, although cheating parties could be detected, the adversary may learn the reconstructed value while the honest parties do not. This was fixed by introducing the concept of

Function $f_{\mathsf{sh}}^{d,\mathsf{perf}}(s)$

Implicit input: $\mathcal{Q} \subseteq \mathcal{P}$.

1. Sample a random polynomial $g(x) \in \mathbb{F}[x]$ of degree at most d such that $g(0) = s$, where $\mathbb{F}$ denotes a finite field with $|\mathbb{F}| > n$.
2. Let $s_i = f(i)$ for $P_i \in \mathcal{Q}$.
3. Output $(s_i)_{P_i \in \mathcal{Q}}$, where s_i is intended for party P_i.

Fig. 7. Generating sharings in the perfectly secure setting

levelled-sharings in Section 5.1.2, which is a method to ensure fairness when reconstructing a shared value. To achieve fairness in our perfect VSS protocol, we use again levelled-sharings in the context of perfect security. The main difference lies on the method that is used to reconstruct individual sharings, since the case of perfect security we can use error correction, instead of the authentication tags developed in Section 5.1.1 for the statistical setting. The details are given below.

5.3.1 Secret Sharing Like in Section 5.1.1, we use Shamir secret sharing. However, unlike the statistical setting, we do not need to authenticate the shares in order to guarantee reconstruction. Instead, we can rely on the error correction properties of Shamir secret sharing, as we show below.

The function $f_{\mathsf{sh}}^{d,\mathsf{perf}}(s)$ that produces sharings of a secret s, which is analogous to $f_{\mathsf{sh}}^{d,\mathsf{stat}}$ in Section 5.1.1, is described in Fig. 7. We denote by $[s]_d$ the output of $f_{\mathsf{sh}}^{d,\mathsf{perf}}(s)$ from an ideal functionality. To reconstruct a secret $[s]_d$ which is d-shared among a set of parties $\mathcal{Q}$ (where $|\mathcal{Q}| = n'$ and $3t'_a + 2t'_p + t'_f < n'$), the protocol $\pi_{\mathsf{PerfRec}}^{d}$ (Fig. 8) is used in the perfect setting. As a basic building block for this protocol we use a Reed-Solomon decoding algorithm $\pi_{\mathsf{RSDec}}(d, W)$ that takes as input a vector W of shares where some of these may be incorrect, and either removes the errors if there are at most $(|W| - d - 1)/2$ of them, or produces $\perp$ (abort) if there are more than $(|W| - d - 1)/2$ errors. This can be instantiated for instance by Berlekamp-Welch algorithm [BW].

The following lemma analyzes correctness of Protocol $\pi_{\mathsf{PerfRec}}^{d}$.

Lemma 7. *Suppose parties in $\mathcal{Q}$ participate in $\pi_{\mathsf{PerfRec}}^{d}$ using the shares computed by $[s]_d$, where $d \leq \lceil n'/3 \rceil - 1$. Then $\pi_{\mathsf{PerfRec}}^{d}$ either outputs the right secret s or $(\perp, \mathcal{C})$ such that $|\mathcal{C}| \geq 1$.*

Proof. It follows directly from the properties of π_{RSDec} that $\pi_{\mathsf{PerfRec}}^{d}$ either produces the right secret s, or $\perp$ together with a set $\mathcal{C}$. It suffices to show that when $|\mathcal{C}| = 0$ holds, then $\pi_{\mathsf{PerfRec}}^{d}$ must result in an output different to $\perp$. This follows from the fact that in such a case $|W| = |\mathcal{Q}| - |\mathcal{C}| = |\mathcal{Q}| = n'$, so $(|W| - d - 1)/2 = (n' - d - 1)/2$. It can be checked that $d \leq \lceil n'/3 \rceil - 1$ implies that the quantity above is lower bounded by $\lceil n'/3 \rceil - 1$,which is bigger than the maximum number of errors t'_a and therefore error-correction succeeds. □

Protocol $\pi^{d}_{\mathsf{PerfRec}}$

Input: A shared value $[s]_d$ among a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ where $3t'_a + 2t'_p + t'_f < n'$.
Output: Secret s' or $\perp$ with a set $\mathcal{C} \subseteq \mathcal{Q}$ of identified corrupt parties (either fail-stop or actively corrupt).
Network Model: Broadcast can be realized using efficient broadcast protocol of [AFM99] that is secure against dynamic adversaries.
Building Block: Decoding algorithm $\pi_{\mathsf{RSDec}}(d, W)$ that takes as input a vector W of shares where some of these may be incorrect, and either removes the errors if there are at most $(|W| - d - 1)/2$ of them, or produces $\perp$ (abort) if there are more than $(|W| - d - 1)/2$ errors.

1. Each P_i broadcasts its share s_i. Let $\mathcal{C} \subseteq \mathcal{Q}$ be the set of parties who did not send s_i. Let W denote the vector constituting the set of values s_k where $P_k \in \mathcal{Q} \setminus \mathcal{C}$.
2. Execute $\pi_{\mathsf{RSDec}}(d, W)$. If the output is $s \neq \perp$, then output s. Else, output $\perp$ and the set $\mathcal{C}$.

Fig. 8. Protocol to reconstruct a d-shared secret in the perfect setting

5.3.2 Levelled-Secret Sharing. In the perfect setting, we use the function $f^{\alpha,\beta}_{\mathsf{pLevSh}}(v)$ defined in Figure 9 to generate (α, β)-levelled sharing of a secret s. This function is analogous to the function $f^{\alpha,\beta}_{\mathsf{sLevSh}}$ from Section 5.1.2 in the statistical setting, with the only difference being that the function $f^{d,\mathsf{perf}}_{\mathsf{sh}}$ is used to produce the individual sharings, instead of $f^{d,\mathsf{stat}}_{\mathsf{sh}}$. We denote by $\langle s \rangle^{\alpha,\beta}$ the output of $f^{\alpha,\beta}_{\mathsf{pLevSh}}(s)$ produced by an ideal functionality.

To reconstruct a (α, β)-levelled shared secret s that has been shared among using parties in $\mathcal{Q}$ according to $f^{d,\mathsf{perf}}_{\mathsf{sh}}(s)$ we use Protocol $\pi^{\alpha,\beta}_{\mathsf{pLevRec}}$ from Fig. 10. This protocol is an straightforward adaptation of Protocol $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$ from Section 5.1.2 to the perfect setting, whose only difference with respect to $\pi^{\alpha,\beta}_{\mathsf{sLevRec}}$ is the fact that error correction, via Protocol $\pi^{d}_{\mathsf{PerfRec}}$ from Fig. 8 is used to reconstruct individual sharings.

We prove the following useful lemmas regarding $f^{\alpha,\beta}_{\mathsf{pLevSh}}(s)$ and Protocol $\pi^{\alpha,\beta}_{\mathsf{pLevRec}}$.

Lemma 8. *Suppose $\langle s \rangle^{\alpha,\beta} \leftarrow f^{\alpha,\beta}_{\mathsf{pLevSh}}(s)$ is computed among parties in $\mathcal{Q}$. Then if $t'_a + t'_p \leq \alpha$, s is perfectly hidden from the adversary.*

Proof. Since the adversary has access only to the shares received on behalf of $t'_a + t'_p \leq \alpha$ parties, it follows from property of Shamir secret sharing that he has no information about the summand s_α which is α-shared. Consequently, s remains perfectly hidden from the adversary. □

Function $f^{\alpha,\beta}_{\mathsf{pLevSh}}(s)$

Implicit input: $\mathcal{Q} \subseteq \mathcal{P}$.
Output: (α,β)-levelled-sharing of s denoted by $\langle s \rangle^{\alpha,\beta}$.
Building Block: $f^{d,\mathsf{perf}}_{\mathsf{sh}}(\cdot)$ (Fig. 7)

1. Sample a random elements $s_\alpha, \ldots, s_\beta \in \mathbb{F}$ such that $\sum_{d=\beta}^{\alpha} s_d = s$
2. For $d = \beta, \ldots, \alpha$, call $[s_d]_d = f^{d,\mathsf{perf}}_{\mathsf{sh}}(s_d)$ using the set $\mathcal{Q}$.
3. Output $([s_\beta]_\beta, \ldots, [s_\alpha]_\alpha)$, where the i-th entry in each $[s_d]_d$ is intended for party P_i.

Fig. 9. Function to compute levelled-secret sharing in perfect setting

Protocol $\pi^{\alpha,\beta}_{\mathsf{pLevRec}}$

Input: A shared value $\langle s \rangle^{\alpha,\beta}$ and a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ where $3t'_a + 2t'_p + t'_f < n'$.
Output: Secret s' or $\perp$ with set $\mathcal{C}$ of identified corrupt parties (either fail-stop or actively corrupt).
Building Block: Protocol $\pi^{d}_{\mathsf{PerfRec}}$ (Fig 8)

1. For $d = \alpha, \ldots, \beta$ each party $P_i \in \mathcal{Q}$ does the following.
   (a) Call $\pi^{d}_{\mathsf{PerfRec}}([s_d]_d)$.
   (b) If the output is s_d, then continue. Else, terminate and output the output of $\pi^{d}_{\mathsf{PerfRec}}([s_d]_d)$ i.e. $(\perp, \mathcal{C})$.
2. Output $s' = s_\alpha + \cdots + s_\beta$.

Fig. 10. Protocol to reconstruct levelled-shared secret in perfect setting

Lemma 9. *Suppose parties in $\mathcal{Q}$ participate in $\pi^{\alpha,\beta}_{\mathsf{pLevRec}}$ using input $\langle v \rangle^{\alpha,\beta}$ computed by $f^{\alpha,\beta}_{\mathsf{pLevSh}}(v)$. Then the following holds:*

(i) Correctness: *Each honest P_i outputs either $s' = v$ or $(\perp, \mathcal{C})$ with $|\mathcal{C}| \geq 1$.*
(ii) Fairness: *If a dynamic adversary (with $3t'_a + 2t'_p + t'_f < n'$) disrupts the reconstruction of s_j ($j \geq 2$), then it does not learn s_{j-1}.*

Proof. Correctness follows directly from the correctness of $\pi^{d}_{\mathsf{PerfRec}}$ (Lemma 7). We present the argument for fairness below.

Suppose the reconstruction of s_j is disrupted. Let $|W| \geq n' - t'_a - t'_f + r$ shares be broadcast during reconstruction, which includes the r shares that were tampered on behalf of r actively corrupt parties. It follows from the correctness of π_{RSDec} used in $\pi^{d}_{\mathsf{PerfRec}}$ that reconstruction of s_j would result in $\perp$ only if there are more than $(|W| - j - 1)/2$ errors, that is, if $|W| \leq j + 2r$, or $n' - t'_a - t'_f + r \leq j + 2r$. Recall that $2t'_a + 2t'_p < n' - t'_a - t'_f$ (implied by $3t'_a + 2t'_p + t'_f < n'$). We can

Protocol π^{d}_{BGW}

Input: A value s from the dealer P_1, and a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ such that the number of active, passive and fail-stop corrupt parties in this set, t'_a, t'_p and t'_f respectively, satisfy $3t'_a + 2t'_p + t'_f < n' := |\mathcal{Q}|$.
Output: Either "disqualified" (indicating that P_1 is disqualified) or $[s]_d$ (i.e the output of $f^{d,\mathsf{perf}}_{\mathsf{sh}}(s)$)

1. P_1 chooses a bivariate polynomial $f^d(x, y)$ of degree d in each variable with $f^d(0, 0) = s$. P_1 sends the polynomial $f_i(x) = f^d(x, i)$ and $g_i(y) = f^d(i, y)$ to P_i $(i = 1, \ldots, n)$.
2. Each pair of parties (P_i, P_j) exchange their cross-over points and check for inconsistencies (i.e whether $f_i(j) \stackrel{?}{=} g_j(i)$ and $f_j(i) \stackrel{?}{=} g_i(j)$)
3. In case of any inconsistencies, the parties broadcast a complaint to the dealer P_1 including the relevant cross-over points.
4. P_1 resolves the conflict between a pair, say (P_i, P_j) by broadcasting the relevant cross-over point w.r.t. which the complaint was made. Corresponding to the *unhappy* party (whose broadcast was inconsistent with the value broadcast by P_1), say P_i, P_1 is supposed to broadcast $f_i(x)$ and $g_i(y)$. Each party checks if these polynomials broadcast by P_1 are consistent with the ones they possess. If not, they broadcast a complaint accusing P_1.
5. If there are more than $\lceil n'/3 \rceil - 1$ accusations against P_1, parties output disqualified and stop.
6. If P_i was unhappy, it sets its polynomials $f_i(x), g_i(y)$ as the ones broadcast by P_1 during complaint resolution Else, P_i uses the polynomials sent by P_1 privately in the beginning of the protocol. Let $s_i = f_i(0)$ denote the respective share of P_i.

Fig. 11. An adaptation of the BGW VSS protocol [BGW88,Dwo90,DDWY93]

therefore infer that $2t'_a + 2t'_p < j + r$, so $t'_a + t'_p < (j + r)/2$, which means that the adversary has access to $\frac{j+r}{2} - 1$ shares at most.

If $r \leq j$, then $\frac{j+r}{2} - 1 \leq j - 1$, so the adversary learns at most $j - 1$ shares, which leak no information about s_{j-1}. It is then left to analyze the case $r > j$, in which it holds that

$$r \leq t'_a \leq \frac{2t'_p + 2t'_a}{2} \leq \frac{n' - t'_a - t'_f - 1}{2} \leq \frac{|W| - r - 1}{2} < \frac{|W| - j - 1}{2}.$$

This implies that $|W| > 2r + j$, which is a contradiction as we assumed above that $|W| \leq 2r + j$. □

5.3.3 Perfectly Secure VSS with Fair Reconstruction We present our protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ for perfect VSS with fair reconstruction in Figures 12 and 13. In a nutshell, our protocol is obtained by using the BGW VSS protocol

Protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$, sharing phase

Let the set of participants $\mathcal{Q}$ be initialized to $\mathcal{P}$, $n' = n, t'_a = t_a, t'_p = t_p, t'_f = t_f$. The *sharing protocol* involving a dealer P_1 with secret s proceeds as follows:

1. P_1 samples random elements $s_1, s_2 \ldots s_{\lceil n'/2 \rceil - 1} \in \mathbb{F}$ such that $\sum_{d=1}^{\lceil n'/2 \rceil - 1} s_d = s$.
2. For each d from 1 to $\lceil n'/2 \rceil - 1$, the parties in $\mathcal{Q}$ run $\pi^d_{\mathsf{BGW}}(s_d)$ and proceed as follows:
   - If some party P_j is detected to fail-stop during the execution of $\pi^d_{\mathsf{BGW}}(s_d)$, then the parties set $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{P_j\}$ and re-run the sharing protocol from the beginning.
   - Else, if the dealer has been disqualified as the output of π_{BGW}, parties output disqualified and stop.
   - Else, parties get $[s_d]_d \leftarrow \pi^d_{\mathsf{BGW}}(s_d)$.
3. Parties output the levelled-sharings

$$\langle s \rangle^{\lceil n'/2 \rceil - 1, 1} = ([s_1]_1, \ldots, [s_{\lceil n'/2 \rceil - 1}]_{\lceil n'/2 \rceil - 1}).$$

Fig. 12. Sharing phase of our protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ for Verifiable Secret Sharing in the perfect setting

[BGW88,Dwo90,DDWY93] as a building block to instantiate the functionality $f^{d,\mathsf{perf}}_{\mathsf{sh}}$ from Section 5.1.1, and then, for the reconstruction phase, Protocol $\pi^{\alpha,\beta}_{\mathsf{pLevRec}}$ is used to reconstruct the levelled sharing. Our adaptation of the BGW VSS protocol appears in Protocol π^d_{BGW} in Fig. 11. As usual, the protocol also takes as input a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ such that the number of active, passive and fail-stop corrupt parties in this set, t'_a, t'_p and t'_f respectively, satisfy $3t'_a + 2t'_p + t'_f < n' := |\mathcal{Q}|$. The protocol guarantees that, on input s from a dealer $P_i \in \mathcal{Q}$, either the parties in $\mathcal{Q}$ obtain consistent shares $[s]_d$, or the dealer is detected as corrupt and disqualified.

We now analyze the properties of the VSS protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ from Figures 12 and 13. First, the privacy of the secret s at the end of sharing phase holds since the adversary cannot learn any information about $s_{\lceil n'/2 \rceil - 1}$ at the end of sharing protocol. This directly follows from the fact that the adversary has access to the shares of at most $\lceil n'/2 \rceil - 1$ parties since $\lceil n'/2 \rceil - 1$ is the maximum value of $t'_a + t'_p$ subject to $3t'_a + 2t'_p + t'_f < n'$. Next, it is easy to check that the correctness of Protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ holds due to correctness of $\pi^d_{\mathsf{PerfRec}}(\cdot)$ (Lemma 7), that is, either the output of the reconstruction phase is the right secret s, or $\perp$.

Lastly we analyze fairness i.e. whether it is possible for the adversary to learn the secret s shared by some honest dealer while the honest parties do not. To this end, suppose reconstruction of s_j fails. It follows from Lemma 9 that, if $j \geq 2$, then the adversary does not learn s_{j-1}, which means it does not learn s as s_{j-1} is a random mask required to reconstruct s. On the other hand, if $j = 1$,

Protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$, reconstruction phase

The fair reconstruction protocol of the VSS proceeds as follows:

1. For each d from $\lceil n'/2 \rceil - 1$ down to 1, the d-shared value s_d can be reconstructed using $\pi^{d}_{\mathsf{PerfRec}}([s_d]_d)$ (Figure 10). If it returns $s'_d \neq \perp$, the parties continue to reconstruction of s_{d-1}. Else they output $\perp$ and terminate.
2. If reconstruction of each among $s_1, s_2 \ldots s_{\lceil n'/2 \rceil - 1}$ is successful, the parties output the secret $s = \sum_{d=1}^{\lceil n'/2 \rceil - 1} s'_d = s'$.

Fig. 13. Reconstruction phase of our protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ for Verifiable Secret Sharing in the perfect setting

then the proof of Lemma 9 shows that $2(t'_a + t'_p) < j + t'_a$, which implies that $t'_a + 2t'_p < j = 1$. From this we see that $t'_a = t'_p = 0$, so in this case fairness is trivial as such an adversary would not have access to any of the messages sent during $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$. This is because the communication throughout $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ is only over pairwise-private channels (recall that broadcast in the perfect setting is realized by adapting standard broadcast protocols that use pairwise-private channels), thereby an adversary with $t'_a = t'_p = 0$ would not receive any message. This completes the description and analysis of the perfect VSS protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ with fair reconstruction against dynamic adversary. This is captured in the following theorem.

Theorem 8. *The protocol $\pi^{\mathsf{perf}}_{\mathsf{VSS}}$ instantiates the fair VSS functionality with perfect security against the adversary $\mathcal{A}^{\mathsf{perf}}$.*

6 General Mixed Adversaries

In [BFH+08] and [HMZ08], SFE and MPC against general mixed adversaries was studied. A general mixed adversary may choose to actively, respectively passively, respectively fail corrupt players in three different subsets, where the triple of subsets must be chosen from a family of triples known as an adversary structure. Combinatorial characterizations are given of the adversary structures that allow for SFE and reactive MPC, with perfect security in [BFH+08] and statistical in [HMZ08].

Since the actual triple (corruption strategy) chosen by the adversary is not given to the protocol, this model also covers the dynamic adversary model we consider here. In a nutshell, our model is the general mixed adversary model, where the adversary is limited to adversary structures described only by subset sizes t_a, t_p and t_f.

The set of players is called P. An adversary structure is a family of triples of subsets of P, $\mathcal{A} = \{(A, E, F)\}$, where the semantics is that the adversary may choose to corrupt players in A actively, players in E passively and fail corrupt

players in F. $\mathcal{A}$ must satisfy some natural monotonicity conditions, coming from the fact that if a subset is corruptible, the adversary could always choose to corrupt any smaller subset. Also, it is assumed that for any triple $A \subseteq E, A \subseteq F$. This is done in [BFH+08] and [HMZ08] to simplify notation, the idea is that an actively corrupt player can behave as if he was passively or fail corrupted.

We will define a threshold adversary structure $\mathcal{T}$ to be one where membership of a triple (A, E, F) in $\mathcal{T}$ can be decided based only on the sizes of A, E and F. Let us define $E' = E \setminus A, F' = F \setminus A$ as the sets that are only passively, resp. only fail corrupted. To get the connection to our parameters t_a, t_p and t_f, we define $Sizes(\mathcal{T})$ to be the family of triples (t_a, t_p, t_f) that occur as sizes of sets (A, E', F') induced by some triple $(A, E, F) \in \mathcal{T}$.

The type of question we ask in this paper can now be rephrased as: given a threshold adversary structure $\mathcal{T}$, what conditions must $Sizes(\mathcal{T})$ satisfy to allow for SFE and MPC?

6.1 Statistical Security

In [HMZ08] two conditions are given on an adversary structure $\mathcal{A}$.

$$C_2 : \forall (A_1, E_1, F_1), (A_2, E_2, F_2) \in \mathcal{A} : \ E_1 \cup E_2 \cup (F_1 \cap F_2) \neq P$$

$$C_1 : \forall (A_1, E_1, F_1), (A_2, E_2, F_2) \in \mathcal{A} : \ E_1 \cup F_2 \neq P$$

It is shown that SFE against $\mathcal{A}$ is possible with statistical security if and only if C_2 is satisfied, and reactive MPC is possible if and only if both C_1 and C_2 are satisfied.

Recall that we defined E_i' and F_i' to be the sets that are only passively resp. only fail corrupted, then the conditions can be written as $A_1 \cup A_2 \cup E_1' \cup E_2' \cup (F_1 \cap F_2) \neq P$ and $A_1 \cup E_1' \cup A_2 \cup F_2' \neq P$

Note that by monotonicity, we can assume without loss of generality that A_1, A_2 are disjoint, that E_1', E_2' are disjoint, and that one F'-set is contained in the other, as this allows us to achieve the same (family of) subset(s) $E_1 \cup E_2 \cup (F_1 \cap F_2)$ that occur in the condition. The same holds for C_1. Hence for a threshold adversary structure $\mathcal{T}$ where only set sizes matter, we can rewrite the conditions as follows:

$$C_2^{th} : \forall (t_a^1, t_p^1, t_f^1), (t_a^2, t_p^2, t_f^2) \in Sizes(\mathcal{T}) : \ t_a^1 + t_a^2 + t_p^1 + t_p^2 + min(t_f^1, t_f^2) < n$$

$$C_1^{th} : \forall (t_a^1, t_p^1, t_f^1), (t_a^2, t_p^2, t_f^2) \in Sizes(\mathcal{T}) : \ t_a^1 + t_p^1 + t_a^2 + t_f^2 < n$$

These are well defined conditions, but not very useful: we would rather have a criterion that describes what conditions a *single* triple t_a, t_p, t_f must satisfy. It turns out that the C_2 is equivalent to the non-dynamic feasibility bound:

Lemma 10. *A threshold adversary structure $\mathcal{T}$ satisfies C_2 (or, equivalently, C_2^{th}) if and only if*

$$\forall (t_a, t_p, t_f) \in Sizes(\mathcal{T}) : \ 2t_a + 2t_p + t_f < n$$

Proof. Assume first C_2 is satisfied. Then we can let $(t_a, t_p, t_f) = (t_a^1, t_p^1, t_f^1) = (t_a^2, t_p^2, t_f^2)$ in C_2^{th} and then $2t_a + 2t_p + t_f < n$ follows immediately. Conversely, assume that $2t_a^1 + 2t_p^1 + t_f^1 < n$ and $2t_a^2 + 2t_p^2 + t_f^2 < n$. Adding these two inequalities gives

$$t_a^1 + t_a^2 + t_p^1 + t_p^2 + (t_f^1 + t_f^2)/2 < n$$

From this C_2^{th} follows, since $min(t_f^1, t_f^2) \leq (t_f^1 + t_f^2)/2$. □

For C_1, we can define for any threshold adversary structure $\mathcal{T}$, the following two values:

$$R_{\mathcal{T}} = max_{(t_a,t_p,t_f)\in Sizes(\mathcal{T})}(t_a + t_p),\ F_{\mathcal{T}} = max_{(t_a,t_p,t_f)\in Sizes(\mathcal{T})}(t_a + t_f).$$

Intuitively, $R_{\mathcal{T}}$ is the maximal number of player states the adversary can read, while $F_{\mathcal{T}}$ is the maximal number of players the adversary can have abort the protocol. It is now immediate that we have

Lemma 11. *A threshold adversary structure $\mathcal{T}$ satisfies C_1 (or, equivalently, C_1^{th}) if and only if $R_{\mathcal{T}} + F_{\mathcal{T}} < n$*

This means that the threshold structures that allow for reactive MPC are characterized, simply as exactly those that satisfy the above two lemmas.

We now give a more concrete characterization of some special cases that satisfy the conditions, as they can be easier to work with. Suppose $\mathcal{T}$ satisfies both C_1 and C_2. It is clear from C_2 that $R_{\mathcal{T}} < n/2$. So if we do not assume any stronger conditions than C_2, we must have $F_{\mathcal{T}} \leq n/2$. This is exactly the bounds we arrived at earlier, $2t_a + 2t_p + t_f < n$ and $t_a + t_f \leq n/2$ (Theorem 2).

But it is also possible to choose a different tradeoff on the parameters, say we demand $3t_a + 3t_p + t_f < n$, which implies C_2 and $R_{\mathcal{T}} < n/3$. Then we can satisfy C_1 by asking that $t_a + t_f \leq 2n/3$, which essentially allows for more fail corruptions.

In general, we can consider any bound of the form $\alpha t_a + \beta t_p + t_f < n$ where $\alpha, \beta \geq 2$ so C_2 is satisfied. It is now not hard to see that the maximal value of $t_p + t_a$ is less than n/m where $m = min\{\alpha, \beta\}$. So C_1 will be satisfied, if $t_a + t_f \leq n(m-1)/m$. To summarize:

Lemma 12. *A threshold adversary structure $\mathcal{T}$ satisfies C_1 and C_2 if for all $(t_a, t_p, t_f) \in Sizes(\mathcal{T})$ it holds that $\alpha t_a + \beta t_p + t_f < n$ and $t_a + t_f \leq n(m-1)/m$, where α, β, m are numbers such that $\alpha, \beta \geq 2$ and $m = min\{\alpha, \beta\}$.*

6.2 Perfect Security

In [BFH+08] characterizations are given on adversary structures that allow for perfectly secure SFE and MPC. They give the following condition

$$C_{mult} : \forall (A_1, E_1, F_1), (A_2, E_2, F_2), (A_3, E_3, F_3) \in \mathcal{A} : E_1 \cup E_2 \cup A_3 \cup (F_1 \cap F_2 \cap F_3) \neq P$$

and show that this and another condition called C_{nrec} is necessary and sufficient for perfect SFE. However, the C_{nrec} is only needed to ensure that a secret-shared

output can be revealed using levelled secret sharing, such that the reconstruction is fair and if it fails, one can identify at least one corrupt player. Now, in the proof of Theorem 8, we show that this is indeed possible efficiently for the entire parameter range $3t_a + 2t_p + t_f < n$. If this condition is violated, not even non-dynamic protocols exist, so we can ignore the C_{nrec} condition in our threshold case.

We can translate C_{mult} condition to a condition on a threshold adversary structure $\mathcal{T}$ exactly as above, to get

$$C^{th}_{mult} : \forall (t^1_a, t^1_p, t^1_f), (t^2_a, t^2_p, t^2_f), (t^3_a, t^3_p, t^3_f) \in Sizes(\mathcal{T}) : t^1_a + t^2_a + t^3_a + t^1_p + t^2_p + min(t^1_f, t^2_f, t^3_f) < n$$

It is easy to see that C^{th}_{mult} implies the inequality $3t_a + 2t_p + t_f < n$ for all triples in $Sizes(\mathcal{T})$ - by taking all the triples in the condition to be equal. However, the converse is clearly false: one can choose maximal values such that $t^3_a < n/3$ and $t^1_p, t^2_p < n/2$ (by having the other parameters be 0) and this clearly sums to more than n.

So hence G.O.D. perfect dynamic SFE is not possible assuming only $3t_a + 2t_p + t_f < n$. Of course, this is no surprise, as we show in Theorem 3 that perfect dynamic SFE is impossible under this assumption, even if we only require security with abort.

The question then is to find a simpler complete characterization of the $\mathcal{T}$'s that satisfy C^{th}_{mult}. We leave the solution of this for future work, but a partial answer is obtained in the following lemma:

Lemma 13. *$Sizes(\mathcal{T})$ satisfies C^{th}_{mult} if $\forall (t_a, t_p, t_f) \in Sizes(\mathcal{T})$ we have that $\alpha t_a + \beta t_p + t_f < n$ where $\alpha \geq \beta$ and $1/\alpha + 2/\beta \leq 1$.*

Proof. We do the proof by analyzing how we can choose 3 triples from $Sizes(\mathcal{T})$ such that we maximize the sum in the condition. Since the sum contains the summand $min(t^1_f, t^2_f, t^3_f)$ it is optimal to have $t^1_f = t^2_f = t^3_f = m$ for some m. Namely, having some t^i_f be larger than the others would force t^i_a, t^i_p to be smaller without increasing the minimum. Further we should clearly choose maximal t^3_a such that $t^3_a < (n-m)/\alpha$. Now, since $\beta \leq \alpha$ we maximize $t^1_a + t^1_p$ by choosing $t^1_a = 0$ and t^1_p maximal such that $t^1_p < (n-m)/\beta$. We see that then the sum will be less than

$$(n-m)/\alpha + 2(n-m)/\beta + m \leq n$$

by the assumption on α, β. □

Note that it is clear from the proof that if we are after a general inequality of form $\alpha t_a + \beta t_p + t_f < n$, then $1/\alpha + 2/\beta \leq 1$ is the weakest condition we can put on α and β. So in this sense, the lemma is optimal.

We allow a larger number of active corruptions by choosing a smaller α, and it is easy to see that $\alpha = 3$ is the smallest value where the condition can be satisfied, and we get that $3t_a + 3t_p + t_f < n$ is sufficient. However, we may as well consider the simpler $3t_a + t_f < n$ because in this case the "price" of active and passive corruptions is the same and it is better for the adversary to corrupt actively.

However, we can also choose larger values of α, and this will allow β to be smaller than 3. In the limit, we can take $t_a = 0$ and then we can have $\beta = 2$, so we get the condition $2t_p + t_f < n$.

In [BFH+08], they also state the condition C_{rec}.

$$C_{rec} : \forall (A_1, E_1, F_1), (A_2, E_2, F_2), (A_3, E_3, F_3) \in \mathcal{A} : \; E_1 \cup A_2 \cup A_3 \cup (F_2 \cap F_3) \neq P$$

And it is shown that reactive perfect MPC is possible if and only both C_{mult} and C_{rec} are satisified. The threshold version of this is

$$C^{th}_{rec} : \forall (t^1_a, t^1_p, t^1_f), (t^2_a, t^2_p, t^2_f), (t^3_a, t^3_p, t^3_f) \in Sizes(\mathcal{T}) : t^1_a + t^2_a + t^3_a + t^1_p + min(t^2_f, t^3_f) < n$$

We can first easily see that the non-dynamic feasibility condition $3t_a + 2t_p + t_f < n$ is not sufficient to ensure that C^{th}_{rec} is satisfied, not even if we also require $t_f = 0$. We have

Lemma 14. *The requirement $3t_a + 2t_p < n$ does not imply C^{th}_{rec}, for $n \geq 7$.*

Proof. The assumption allows us to choose $(t^1_a, t^1_p, t^1_f), (t^2_a, t^2_p, t^2_f), (t^3_a, t^3_p, t^3_f)$ as $(0, t^1_p, 0), (t^2_a, 0, 0), (t^3_a, 0, 0)$, where t^2_a, t^3_a are maximal such that they are smaller than $n/3$ and t^1_p maximal smaller than $n/2$. It is easy to see that this will violate C^{th}_{rec} when $n \geq 7$. □

We now analyse the C^{th}_{rec} condition for the two special cases we arrived at above, where no passive or no active corruptions are allowed, and we want to derive in each case the weakest possible additional condition we can put so that C_{rec} is also satisfied.

Lemma 15. *Let $\mathcal{T}$ be the threshold adversary structure such that $Sizes(\mathcal{T})$ contains all triples satisfying $t_p = 0$ and $3t_a + t_f < n$. Construct a new structure $\mathcal{S} \subseteq \mathcal{T}$ by selecting from $Sizes(\mathcal{T})$ only those triples satisfying $3t_a + 3/2t_f \leq n$. Then $\mathcal{S}$ satisfies C_{rec} (and C_{mult}). Furthermore, if we do the same with the weaker inequality $3t_a + \delta t_f \leq n$ for $\delta < 3/2$, then C_{rec} is violated.*

Proof. Let us assume an inequality $3t_a + \delta t_f \leq n$ and figure out what δ needs to be to ensure C_{rec}.

The sum in the condition becomes $t^1_a + t^2_a + t^3_a + min(t^2_f, t^3_f)$, and as in the previous lemma, it is optimal to have $t^2_f = t^3_f = m$ for some m, where we know that $m \leq n/\delta$. We can freely maximize t^1_a so it is less than $n/3$ (satisfying our assumption $3t_a + t_f < n$), while the largest value we can have for t^2_a, t^3_a is the maximal value $(n - \delta m)/3$. Now, the sum is less than $n/3 + 2(n - \delta m)/3 + m$, so to get C_{rec} we need

$$n/3 + 2(n - \delta m)/3 + m \leq n$$

which simplifies to

$$(3 - 2\delta)m \leq 0$$

So this implies that if we want to allow fail corruptions at all, the smallest value of δ that will work is $\delta = 3/2$, any smaller value would violate the inequality. □

Lemma 16. *Let $\mathcal{T}$ be the threshold adversary structure such that $Sizes(\mathcal{T})$ contains all triples satisfying $t_a = 0$ and $2t_p + t_f < n$. Construct a new structure $\mathcal{S} \subseteq \mathcal{T}$ by selecting from $Sizes(\mathcal{T})$ only those triples satisfying $2t_p + 2t_f \leq n$. Then $\mathcal{S}$ satisfies C_{rec} (and C_{mult}). Furthermore, if we do the same with the weaker inequality $2t_p + \gamma t_f \leq n$ for $\gamma < 2$, then C_{rec} is violated.*

Proof. It is very easy to see that assuming the inequality $2t_p + 2t_f \leq n$, C_{rec} is satisfied, as the t_a-summands are 0, t_p summands are less than $n/2$ (as we assume $2t_p + t_f < n$) and t_f summands are at most $n/2$. It is also clear that if $\gamma < 2$, we can have $min(t_f^2, t_f^3) \geq n/2$ and C_{rec} fails. □

7 Acknowledgments

Divya Ravi was funded by the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 803096 (SPEC). During his time in Aarhus University, Daniel Escudero was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO).

References

AFM99. Bernd Altmann, Matthias Fitzi, and Ueli M. Maurer. Byzantine agreement secure against general adversaries in the dual failure model. In Prasad Jayanti, editor, *Distributed Computing, 13th International Symposium, Bratislava, Slovak Republic, September 27-29, 1999, Proceedings*, volume 1693 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 1999.

BFH+08. Zuzana Beerliová-Trubíniová, Matthias Fitzi, Martin Hirt, Ueli M. Maurer, and Vassilis Zikas. MPC vs. SFE: perfect security in a unified corruption model. In *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 231–250. Springer, 2008.

BFO12. Eli Ben-Sasson, Serge Fehr, and Rafail Ostrovsky. Near-linear unconditionally-secure multiparty computation with a dishonest minority. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 663–680, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

BGW88. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10, Chicago, IL, USA, May 2–4, 1988. ACM Press.

BTH08. Zuzana Beerliová-Trubíniová and Martin Hirt. Perfectly-secure MPC with linear communication complexity. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 213–230, San Francisco, CA, USA, March 19–21, 2008. Springer, Heidelberg, Germany.

BW. E. R. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent Number 4,633,470. Issued Dec. 1986.

Can00. Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000.

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th FOCS*, pages 383–395, Portland, Oregon, October 21–23, 1985. IEEE Computer Society Press.

DDWY93. Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

DN07. Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 572–590, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany.

Dwo90. Cynthia Dwork. Strong verifiable secret sharing (extended abstract). In *Distributed Algorithms, 4th International Workshop, WDAG '90, Bari, Italy, September 24-26, 1990, Proceedings*, pages 213–227, 1990.

FHM98. Matthias Fitzi, Martin Hirt, and Ueli M. Maurer. Trading correctness for privacy in unconditional multi-party computation (extended abstract). In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 121–136, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany.

GSZ20. Vipul Goyal, Yifan Song, and Chenzhi Zhu. Guaranteed output delivery comes free in honest majority MPC. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020 , Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 618–646. Springer, 2020.

HLM13. Martin Hirt, Christoph Lucas, and Ueli Maurer. A dynamic tradeoff between active and passive corruptions in secure multi-party computation. In *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 203–219. Springer, 2013.

HLMR11. Martin Hirt, Christoph Lucas, Ueli Maurer, and Dominik Raub. Graceful degradation in multi-party computation (extended abstract). In Serge Fehr, editor, *ICITS 11*, volume 6673 of *LNCS*, pages 163–180, Amsterdam, The Netherlands, May 21–24, 2011. Springer, Heidelberg, Germany.

HM20. Martin Hirt and Marta Mularczyk. Efficient MPC with a mixed adversary. *IACR Cryptol. ePrint Arch.*, 2020:356, 2020.

HMZ08. Martin Hirt, Ueli M. Maurer, and Vassilis Zikas. MPC vs. SFE: Unconditional and computational security. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 1–18, Melbourne, Australia, December 7–11, 2008. Springer, Heidelberg, Germany.

IK02. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256, Malaga, Spain, July 8–13, 2002. Springer, Heidelberg, Germany.

PR19. Arpita Patra and Divya Ravi. Beyond honest majority: The round complexity of fair and robust multi-party computation. In *ASIACRYPT (1)*, volume 11921 of *Lecture Notes in Computer Science*, pages 456–487. Springer, 2019.

RB89. Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *21st ACM STOC*, pages 73–85, Seattle, WA, USA, May 15–17, 1989. ACM Press.

A Supplement to Impossibility Results

A.1 Impossibility of [FHM98].

Lemma 17. *Let f' denote a function computing the logical* AND *of two inputs bits x'_1 and x'_2 held by P_1^* and P_2^* respectively. Then no 3-party protocol computing f' among $\{P_1^*, P_2^*, P_3^*\}$ can achieve perfect security with abort against an adversary that corrupts either P_1^* or P_2^* passively or P_3^* actively.*

Proof. Suppose by contradiction, there exists a 3-party protocol π' computing f' that achieves perfect security with abort against an adversary that corrupts either P_1^* or P_2^* passively or P_3^* actively. Let T_{ij} $(1 \leq i < j \leq 3)$ denote the transcript of the private communication between P_i^* and P_j^* and T denote the transcript of the communication via broadcast channels. We consider the following sequence of observations:

- T_{12} and T must be independent of x'_1 and x'_2. If not, consider the first message in T_{12} or T that depends on the input of its sender, say P_1^*. Then, a passively corrupt P_2^* can learn x'_1 irrespective of x'_2 (which should not be allowed as per the ideal realization of f'). Therefore, we can conclude that P_1^*'s output cannot be determined by $\{T_{12}, T\}$ alone and should depend on the messages received from both P_2^* and P_3^* i.e $\{T_{12}, T_{13}, T\}$.
- Consider two executions E_1, E_2 where $(x'_1, x'_2) = (1, 0)$ and $(x'_1, x'_2) = (1, 1)$ respectively. Suppose no one misbehaves in E_1, E_2. Then the protocol must result in all parties obtaining the correct output which is 0 and 1 respectively. Assume that T, T_{12} is identical across E_1, E_2. This is possible based on the above observation.
- Consider an execution E_3 where $(x'_1, x'_2) = (1, 1)$ where an actively corrupt P_3^* replaces his messages to P_1^*, P_2^* with random strings. With some small but non-zero probability, the view of P_1^* may be identical to his view in E_1 resulting in him outputting 0 which violates correctness. Note that this holds even though π' achieves security with abort as P_1^*'s view is identical to E_1 where all parties behave honestly and therefore P_1^* does not output $\perp$.

We have thus arrived at a contradiction to our assumption that π' is secure, completing the proof of Lemma 17. □

A.2 Generalization of Theorem 3.

The proof of Theorem 3 can be extended for $n > 6$ by tweaking the transformation from the n-party protocol π to the 3-party protocol π' in the following manner: P_1^* emulates the role of some set S_1 of $\lceil n/2 \rceil - 1$ parties, P_2^* emulates another disjoint set S_2 of $\lceil n/2 \rceil - 1$ parties and P_3^* emulates the role of the remaining set of parties S_3, which comprises of 2 parties or 1 party (depending on whether n is even or odd respectively). We see that the resulting protocol tolerates either a passively corrupt P_1^*, a passively corrupt P_2^* or an actively corrupt P_3^*, since these

correspond to passively corrupting S_1, S_2, or actively corrupting S_3 [7] respectively, which is allowed by π as per assumption. The rest of the argument remains same as before.

B Missing Proofs from Section 5

Lemma 18 (Lemma 5 restated). *Consider an actively corrupt party P_i and an honest party in $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$ in protocol π^d_{StatRec}. Let s_i be P_i's share in $[s]_d$, and suppose P_i broadcasts $s'_i \neq s_i$ in the first step. Then, with probability at least $1 - \frac{1}{|\mathbb{F}|}$, P_j broadcasts (accuse, P_i) in the accusation round.*

Proof. Begin by writing $s'_i = s_i + \delta_i$, and suppose that P_i broadcasted $m'_{ij} = m_{ij} + \gamma_{ij}$ for some errors $\delta_i, \gamma_{ij} \in \mathbb{F}$ with $\delta_i \neq 0$. The honest P_j checks the equation $m'_{ij} \stackrel{?}{=} \alpha_{ji} \cdot s'_i + \beta_{ji}$, and we claim that this equation is satisfied only with probability $1/|\mathbb{F}|$. To see this observe that by construction $m_{ij} = \alpha_{ji} \cdot s_i + \beta_{ji}$, so, if the equation above is satisfied, we have that $\alpha_{ji} \cdot \delta_i = \gamma_{ij}$, but since $\delta_i \neq 0$, this implies that $\alpha_{ji} = \gamma_{ij}/\delta_i$. Since γ_{ij}/δ_i are chosen by the adversary independently of α_{ji}, which follows from the fact that P_i knows nothing about α_{ji} because the random value β_{ji} masks α_{ji} in m_{ij}, we conclude that the adversary can only satisfy the above equation with probability at most $1/|\mathbb{F}|$. □

Theorem 9 (Theorem 5 restated). *Protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$ evaluates the function f in $O(n)$ rounds with statistical security against $\mathcal{A}^{\mathsf{stat}}$.*

Proof. We model the protocol π_{StatBase} as a functionality $\mathcal{F}_{\mathsf{StatBase}}$ that, on top of receiving the inputs for f from all the parties, receives as input from the adversary a set $\mathcal{F} \subseteq \mathcal{P}$ of fail-stop parties, and computes the functionality $f^{\lceil n'/2 \rceil - 1,1}_{\mathsf{sLevSh}} \circ f$ on the inputs of parties in $\mathcal{P} \setminus \mathcal{F}$, with default inputs for the parties in $\mathcal{F}$. This is the functionality instantiated by the protocol π_{StatBase}. We define a simulator S that interacts with the adversary A, and with an ideal functionality for evaluating f, in such a way that the adversary cannot distinguish, up to a negligible statistical error, whether it is interacting with the actual honest parties in a real execution, or with the simulator in the ideal execution. The simulator is defined as follows:

1. S initializes $\mathcal{Q} := \mathcal{P}$.
2. S emulates the functionality $\mathcal{F}_{\mathsf{StatBase}}$, and it also emulates virtual honest parties. The adversary begins by sending the inputs from the corrupt parties to the functionality $\mathcal{F}_{\mathsf{StatBase}}$, as well as the set $\mathcal{F}$.
3. S sets $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \mathcal{F}$, and then it calls $([\widehat{y_d}]_d)_{d=1}^{\lceil n'/2 \rceil - 1} = f^{\lceil n'/2 \rceil - 1,1}_{\mathsf{sLevSh}}(y')$, where $y' \in \mathbb{F}$ is a random dummy value, and sends the actively and passively corrupted parties in $\mathcal{Q}$ their respective shares and a random element $r' \in \mathbb{F}$.

[7] when $n > 6$, π must be able to tolerate at least two active corruptions which is the maximal size of S_3.

4. The parties then engage in the protocol $\pi_{\mathsf{sLevRec}}^{\lceil n'/2\rceil-1,1}(\langle y'\rangle^{\lceil n'/2\rceil-1,1})$. S emulates this execution as follows: Let $\delta = t'_a + t'_p \leq \lceil n'/2\rceil - 1$, which in particular means that adversary knows the values $\widehat{y}_1, \ldots, \widehat{y}_{\delta-1}$. S emulates the honest parties in the steps corresponding to $d = \lceil n'/2\rceil - 1, \ldots, 1$ in protocol $\pi_{\mathsf{sLevRec}}^{\lceil n'/2\rceil-1,1}(\langle y'\rangle^{\lceil n'/2\rceil-1,1})$ as follows:
   (a) If $\delta < d \leq \lceil n'/2\rceil - 1$, emulate the steps of $\pi_{\mathsf{sLevRec}}^{\lceil n'/2\rceil-1,1}(\langle y'\rangle^{\lceil n'/2\rceil-1,1})$ on behalf of honest parties. Let $\mathcal{F}$ denote the set of fail-stop parties and $\mathcal{A}$ denote the set of corrupt parties who broadcast shares inconsistent with what was returned to them in Step 3. In the 'accusation' round, complain against $P_i \in \mathcal{A}$ on behalf of honest parties in $\mathcal{Q}$. If $n' - |\mathcal{F}| - |\mathcal{A}| \leq d$, then S updates $\mathcal{Q} \leftarrow \mathcal{Q} \setminus (\mathcal{A} \cup \mathcal{F})$, $n' = |\mathcal{Q}|$ and repeats the simulation from step 2 above. Else S continues.
   (b) If $d = \delta$, then S queries the functionality that computes f setting the input of the corrupted parties in $\mathcal{Q}$ to be what S received from the adversary in step 2, and setting the inputs of parties in $\mathcal{P} \setminus \mathcal{Q}$ to default values. S gets the output y, updates $\widehat{y}_\delta \leftarrow y - \sum_{d\neq\delta} \widehat{y}_d + r'$, and emulates the honest parties in the reconstruction of $[\widehat{y}_\delta]_\delta$, modifying the shares from the virtual honest parties so that the reconstructed value matches the updated $\widehat{y}_\delta$. Notice that this is possible since the adversary controls only δ shares.
   (c) If $1 \leq d \leq \delta - 1$, S emulates the honest parties in the reconstruction of $[\widehat{y}_d]_d$. S instructs the functionality to provide output to the honest parties that were not declared fail-stop up to this point.

To see that the real and ideal executions are statistically indistinguishable, we begin by observing that, if step 4.a is reached, then the adversary cannot distinguish between the two executions up to that point. This follows from the fact that S perfectly emulates the real world, except in the following aspects: First, it uses a dummy random value y' for the levelled sharing. However, this is acceptable because, during step 4.a, the adversary does not learn $\widehat{y}_\delta$ (implied by Proposition 2), which perfectly hides the value of $\widehat{y}$ (subsequently, y) in the real execution. Second, the difference between real and ideal execution is the manner in which a corrupt P_i is included in $\mathcal{A}$ - In the former, a corrupt P_i is included in $\mathcal{A}$ if atleast $\lceil \frac{n''+1}{2} \rceil$ parties accuse P_i of failing the mac verification, where $n'' = |\mathcal{Q}| - |\mathcal{F}_1| - |\mathcal{F}_2|$. In the latter, P_i is included in $\mathcal{A}$, if P_i broadcasts share inconsistent with what was received as output of $\mathcal{F}_{\mathsf{StatBase}}$. Indistiguishability follows directly from Lemma 5 and the fact that there are atleast $\lceil \frac{n''+1}{2} \rceil$ honest parties among $\mathcal{Q} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2)$.

Next, we consider the special case when $\delta = 0$ (i.e. $t'_a = t'_p = 0$) and the simulation terminates after step 4.a. In the ideal execution, the adversary learns the dummy random element y' which perfectly emulates the real execution where the adversary learns the masked output $\widehat{y}$ (which perfectly hides the output y, as the adversary with $t'_a = t'_p = 0$ has no information about the random mask).

Given the above, it only remains to look at the case when $\delta \geq 1$ and in which step 4.b (and subsequently 4.c) is reached. From Proposition 1, with overwhelming probability, if this step is reached then it is because all the reconstructed values

$\widehat{y}_{\delta+1}, \ldots, \widehat{y}_{\lceil n'/2 \rceil - 1}$ are correct. In the ideal world, the output y is computed from the inputs of the parties in the current set $\mathcal{Q}$, and the output is provided to all honest parties in $\mathcal{Q}$ that were not signalled as fail-stop. We argue that this also happens in the real world. To this end, it is enough to show that the adversary cannot disrupt the reconstruction of $[\widehat{y}_d]_d$ for $d = \delta, \ldots, 1$, since in this case the honest parties will be able to obtain the output in the real execution. To see this simply note that, from the fault identification property in Proposition 2, if the adversary disrupts the reconstruction of $[\widehat{y}_d]_d$ for some $d = \delta, \ldots, 1$, then a pair of sets $\mathcal{A}, \mathcal{F} \subseteq \mathcal{Q}$ of actively and fail-stop corrupted parties with $|\mathcal{F} \cup \mathcal{A}| \geq n' - d$ is produced. However, this implies that

$$t'_f + t'_a \geq n' - d \geq n' - (t'_a + t'_p) > (t'_f + 2t'_a + 2t'_p) - (t'_a + t'_p) = t'_f + t'_a + t'_p,$$

which is a contradiction.

□

C Some Special Cases

We saw in Sections 3 and 4 multiple impossibilities results for MPC against a dynamic adversary that satisfies $3t_a + 2t_p + t_f < n$ in the perfect setting ($\mathcal{A}^{\mathsf{perf}}$), and $2t_a + 2t_p + t_f < n$ in the statistical setting ($\mathcal{A}^{\mathsf{stat}}$). More specifically, Theorem 2 in Section 3 shows that statistically secure VSS with G.O.D. against $\mathcal{A}^{\mathsf{stat}}$ is impossible. Perfectly secure SFE with abort is impossible against $\mathcal{A}^{\mathsf{perf}}$, even if $t_f = 0$ (Section 4.1) and perfectly secure VSS with G.O.D. against $\mathcal{A}^{\mathsf{perf}}$ is also impossible, even if $t_f = 0$ (Section 4.2).

However, if add some extra assumptions we can show that these impossibility results do not hold anymore. Concretely, we show in Section C.1 below that reactive dynamic MPC with G.O.D. and statistical security is possible, if we introduce the additional assumption $R_{\mathcal{S}} + F_{\mathcal{S}} < n$ (shown to be necessary in Theorem 2). Also, in Sections C.2 and C.3 we show that perfectly secure SFE and VSS with G.O.D. are both possible if we assume either that $t_a = 0$ or $t_p = 0$.

C.1 Reactive MPC with Statistical Security

Theorem 10. *Suppose $\mathcal{S}$ is the set of corruption strategies the adversary can choose from, $R_{\mathcal{S}}$ is the maximal number of player state the adversary can read and $F_{\mathcal{S}}$ is the maximal number of players the adversary can have abort. In the statistical setting, the condition $R_{\mathcal{S}} + F_{\mathcal{S}} < n$ is necessary and sufficient to design a VSS with G.O.D against a dynamic adversary.*

Proof. It follows from Theorem 2 that it is possible to design a VSS with G.O.D against $\mathcal{A}^{\mathsf{stat}}$ only if $R_{\mathcal{S}} + F_{\mathcal{S}} < n$, implying that this additional condition is necessary. We prove its sufficiency by describing a VSS protocol below that achieves G.O.D against $\mathcal{A}^{\mathsf{stat}}$, when $R_{\mathcal{S}} + F_{\mathcal{S}} < n$ holds.

For the sharing protocol we use our SFE protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$ from Fig. 6 in Section 5.1.3 to evaluate the function $f^{R_{\mathcal{S}},\mathsf{stat}}_{\mathsf{sh}}(\cdot)$, where the dealer inputs its secret

value s, producing sharings $[s]_{R_S}$. Notice that this preserves privacy as R_S is the maximal number of player states that $\mathcal{A}^{\mathsf{stat}}$ can read. To reconstruct, the parties use protocol $\pi^{R_S}_{\mathsf{StatRec}}$ from Fig. 3. As we saw in Proposition 1, either the parties succeed in the reconstruction, or they fail while outputting a set of size at least $n - R_S$ containing only active and fail-stop parties. However, the latter cannot happen since $n - R_S > F_S$, so the only possibility remaining is that reconstruction always succeeds. □

Since VSS and SFE together imply general reactive MPC by verifiably secret-sharing the intermediate states [HMZ08], we obtain the following result for reactive MPC.

Theorem 11. *Suppose $\mathcal{S}$ is the set of corruption strategies the adversary can choose from, R_S is the maximal number of player state the adversary can read and F_S is the maximal number of players the adversary can have abort. In the statistical setting, reactive MPC with G.O.D. is possible against a dynamic adversary who is allowed to choose a corruption strategy such that $R_S + F_S < n$.*

Lastly, we note that it is easy to view the above construction in terms of (t_a, t_p, t_f) as well. Specifically, the above construction can be used to show that $2t_a + 2t_f \leq n$ is sufficient for VSS and reactive MPC with G.O.D. In more detail, the sharing protocol, $\pi^{\mathsf{stat}}_{\mathsf{god}}$ can be used to generate sharings $[s]_{\lceil n/2 \rceil - 1}$ with threshold $\lceil n/2 \rceil - 1$ (maintains privacy as dynamic adversary needs to respect $2t_a + 2t_p + t_f < n$); while $\pi^{\lceil n/2 \rceil - 1}_{\mathsf{StatRec}}([s]_{\lceil n/2 \rceil - 1})$ constitutes the reconstruction protocol. It is easy to see that the reconstruction always succeeds (if it fails, a set of $n - (\lceil n/2 \rceil - 1) = \lfloor n/2 \rfloor + 1$ parties must be output comprising of only active and fail-stop parties which is not possible as $t_a + t_f \leq n/2$ as per the assumption).

C.2 Perfectly Secure SFE

Notation. The definitions and the protocols in this section are defined with respect to a set of parties $\mathcal{Q} \subseteq \mathcal{P}$. Looking ahead, this set of parties denote the subset of parties in $\mathcal{P}$ that remain after some of the corrupt parties (fail-stop or actively corrupt) are identified and eliminated. Importantly, the bound $3t'_a + 2t'_p + t'_f < n'$ also holds for the set $\mathcal{Q}$, where $|\mathcal{Q}| = n' \leq n$.

C.2.1 Upper Bound ($t_a = 0$). In this section, we consider an adversary $\mathcal{A}^{(t_p,t_f)}$ with $t_a = 0$. In particular, $\mathcal{A}^{(t_p,t_f)}$ can choose (t_p, t_f) subject to $2t_p + t_f < n$ (as opposed to $3t_a + 2t_p + t_f < n$). Unlike the case of $t_f = 0$ of Section 4.1, it turns out that it is feasible to design perfect dynamic SFE with G.O.D against $\mathcal{A}^{(t_p,t_f)}$.

The perfectly-secure protocol $\pi^{t_a=0}_{\mathsf{god}}$ presented in Fig 14 achieves G.O.D. against $\mathcal{A}^{(t_p,t_f)}$. Similar to our protocol $\pi^{\mathsf{stat}}_{\mathsf{god}}$ (Fig. 6), the protocol proceeds in two phases – In Phase 1, a perfectly-secure MPC protocol ϕ that achieves G.O.D against $t < n/2$ *passive* corruptions is executed. ϕ can be instantiated using existing

protocols like the one from [DN07]. Accounting for the fact that these protocols achieve G.O.D. only against a passive adversary and attain security with abort when there are fail-stop corruptions, we do not invoke ϕ to compute the desired function f directly. Instead, ϕ is used to compute an $(\lceil n/2 \rceil - 1, 1)$-levelled sharing of the output of f. Privacy of the levelled-sharing (Lemma 8) ensures that when ϕ results in honest parties obtaining $\perp$, the adversary who learns the internal state of $t_p \leq \lceil n/2 \rceil - 1$ parties does not learn the output either. The fail-corrupt parties who crashed during ϕ are subsequently eliminated and Phase 1 is re-run until it is successful. After $(n-1)$ runs in the worst case (when one fail-corrupt party is eliminated in each run), the parties proceed to Phase 2, where they attempt to reconstruct the $(\lceil n/2 \rceil - 1, 1)$-levelled shared output. If the reconstruction fails, then the parties restart from Phase 1 upon eliminating the fail-corrupt parties who crashed during Phase 2. This completes the high-level description of the protocol $\pi_{\mathsf{god}}^{t_a=0}$. The formal description of $\pi_{\mathsf{god}}^{t_a=0}$, analysis of its correctness and security appears below.

Lemma 19. *Protocol $\pi_{\mathsf{god}}^{t_a=0}$ computes the correct output*

Proof. The correctness of $\pi_{\mathsf{god}}^{t_a=0}$ follows directly from correctness of ϕ and correctness of $\pi_{\mathsf{pLevRec}}^{\lceil n'/2 \rceil - 1, 1}(\cdot)$ (Lemma 9). □

We state the formal theorem below.

Theorem 12. *There exists a perfectly-secure protocol that achieves G.O.D. against an adversary who can choose (t_p, t_f) such that $2t_p + t_f < n$.*

We only present an informal argument for this theorem. First, it is easy to check from the protocol steps that the honest parties will receive the output after at most $(n-1)$ re-runs (in the worst case, Phase 1 will eliminate one fail-corrupt party in each run). Next, we argue that $\mathcal{A}^{(t_p,t_f)}$ will receive a unique output (identical to the honest parties). He learns no information about the output corresponding to the unsuccessful runs (in which $n' \leq n$ parties participate) due to the following reasons: **(a)** Suppose Phase 1 is unsuccessful. Note that $\mathcal{A}^{(t_p,t_f)}$ can receive only upto $t'_p \leq \lceil n'/2 \rceil - 1$ shares of the output. It follows from Lemma 8 that this information perfectly hides the output y which is $(\lceil n'/2 \rceil - 1, 1)$-levelled shared. **(b)** Suppose that Phase 2 is unsuccessful due to failure in reconstruction of summand y_j $(j \geq 2)$. Then it follows from the property of fairness of $\pi_{\mathsf{pLevRec}}^{\lceil n'/2 \rceil - 1, 1}$ (Lemma 9) that the adversary cannot learn y_{j-1} and subsequently y. Lastly, suppose that there is failure in reconstructing summand y_1. This would occur when $1 \geq n' - t'_f > 2t'_p$ (as $2t'_p + t'_f < n'$ holds), which implies $t'_p = 0$ and $t'_f = n' - 1$. In this case, it follows trivially that the adversary has no information about the output. This is because an adversary with $t'_p = 0$ will not have access to any message sent during $\pi_{\mathsf{god}}^{t_a=0}$, as all communication throughout $\pi_{\mathsf{god}}^{t_a=0}$ occurs over pairwise-private channels (recall that broadcast in the perfect setting is realized by adapting standard broadcast protocols that use pairwise-private channels). This completes the intuition.

Protocol $\pi_{\mathsf{god}}^{t_a=0}$

Inputs: Each party P_i participates with input x_i ($i \in [n]$)
Output: $f(x_1, \ldots, x_n)$
Building Blocks: - A perfectly secure MPC protocol ϕ achieving G.O.D against $t < n/2$ *passive* corruptions (instantiated by [DN07]). ϕ achieves *security with abort* in the presence of $\mathcal{A}^{(t_p,t_f)}$ respecting $2t_p + t_f < n$; Function $f_{\mathsf{pLevSh}}^{\alpha,\beta}$ (Fig 9) and Protocol $\pi_{\mathsf{pLevRec}}^{\alpha,\beta}$ (Fig 10)

Initialization: $\mathcal{Q} = \mathcal{P}$, $\mathcal{C} = \emptyset$, $n' = n$, $t'_f = t_f, t'_p = t_p$, $t'_a = t_a = 0$.

Phase 1: Each $P_i \in \mathcal{Q}$ does the following:
- Participate in an execution of ϕ with input x_i to compute $f_{\mathsf{pLevSh}}^{\lceil n'/2 \rceil - 1,1}(y)$, where $y = f(x_1, \ldots, x_n)$ (default inputs used for parties in $\mathcal{C}$). [a]
- If a set of parties S with $|S| \geq 1$ crash during ϕ, update $\mathcal{C} = \mathcal{C} \cup S$, $\mathcal{Q} = \mathcal{Q} \setminus S, t'_f = t'_f - |S|$ and $n' = n' - |S|$. Re-run Phase 1.
- Else, proceed to Phase 2.

Phase 2: Let $\langle y \rangle^{\lceil n'/2 \rceil - 1,1}$ denote the output of Phase 1. The parties in $\mathcal{Q}$ do the following:
- Run $\pi_{\mathsf{pLevRec}}^{\lceil n'/2 \rceil - 1,1}(\langle y \rangle^{\lceil n'/2 \rceil - 1,1})$ to reconstruct the $(\lceil n'/2 \rceil - 1, 1)$-levelled shared output. [b]
- Suppose it outputs $y' \neq \perp$, output y'.
- Else, suppose it outputs $(\perp, S)$. Then, update $\mathcal{C} = \mathcal{C} \cup S$, $\mathcal{Q} = \mathcal{Q} \setminus S, t'_f = t'_f - |S|$ and $n' = n' - |S|$. Restart from Phase 1.

[a] If $n' \leq 2$, ϕ is used to compute f directly as there are no passive corruptions
[b] Since active corruptions are not present, the following simplified variant of $\pi_{\mathsf{pLevRec}}^{\lceil n'/2 \rceil - 1,1}(\langle y \rangle^{\lceil n'/2 \rceil - 1,1})$ can be used alternately - During reconstruction of summand y_δ ($\delta \in [1, \lceil n'/2 \rceil - 1]$), if at least $\delta + 1$ parties broadcast their shares, interpolate a polynomial $A(x)$ of degree δ using the shares and compute $y_\delta = A(0)$. Else output $(\perp, \mathcal{C})$ where $\mathcal{C}$ comprises of the parties who crashed.

Fig. 14. Perfect SFE with G.O.D against dynamic adversary with $t_a = 0$

C.2.2 Upper Bound ($t_p = 0$). In this section, we consider an adversary $\mathcal{A}^{(t_a,t_f)}$ with $t_p = 0$, which in particular means that $\mathcal{A}^{(t_a,t_f)}$ can choose (t_a, t_f) subject to $3t_a + t_f < n$ (as opposed to $3t_a + 2t_p + t_f < n$). In this case it is also feasible to design perfect dynamic SFE with G.O.D. against $\mathcal{A}^{(t_a,t_f)}$, as we now show.

The perfectly-secure protocol $\pi_{\mathsf{god}}^{t_p=0}$ presented in Figure 15 achieves G.O.D against $\mathcal{A}^{(t_a,t_f)}$. At a high-level, $\pi_{\mathsf{god}}^{t_p=0}$ proceeds similar to $\pi_{\mathsf{god}}^{t_a=0}$ (Section C.2.1) i.e. it comprises of two phases such that a sharing of the output is generated in Phase 1 which is subsequently reconstructed in Phase 2. However, it differs in two main aspects – First, Phase 1 involves a protocol ψ that achieves G.O.D against $t < n/3$ *active* corruptions. ψ can be instantiated using existing protocols like [BTH08]. By aborting when a fail-stop party is detected, this protocol achieves security with abort in the presence of fail-stop corruptions and G.O.D. under active-only corruptions. The second distinction is with respect to the thresholds of the levelled sharing of the output. ψ is used to compute $(\lceil n/3 \rceil - 1, 1)$-levelled sharing of the output. Phase 2 involves reconstruction of the levelled-shared output. The fairness of levelled-sharing (Lemma 9) ensures that the adversary does not obtain multiple evaluations of f. This complete the high-level description of $\pi_{\mathsf{god}}^{t_p=0}$. The formal description of $\pi_{\mathsf{god}}^{t_p=0}$ appears in Fig. 15, and the analysis of its correctness and security appears below.

Lemma 20. *Protocol $\pi_{\mathsf{god}}^{t_p=0}$ computes the correct output.*

Proof. Correctness of $\pi_{\mathsf{god}}^{t_p=0}$ follows directly from correctness of ψ and correctness of $\pi_{\mathsf{pLevRec}}^{\lceil n'/3 \rceil - 1,1}(\cdot)$ (Lemma 9). □

We state the formal theorem below.

Theorem 13. *There exists a perfectly-secure protocol that achieves G.O.D against an adversary who can choose (t_a, t_f) such that $3t_a + t_f < n$.*

We provide some intuition for the validity of this theorem below. First, it is easy to check from the protocol steps that the honest parties will receive the output after atmost $(n-1)$ re-runs (in the worst case, Phase 1 will eliminate one fail-corrupt party in each run). Next, we argue that $\mathcal{A}^{(t_a,t_f)}$ will receive a unique output (identical to the honest parties). He learns no information about the output corresponding to the unsuccessful runs (in which $n' \leq n$ parties participate) due to the following reasons: **(a)** Suppose Phase 1 is unsuccessful. Note that $\mathcal{A}^{(t_a,t_f)}$ can receive only upto $t'_a \leq \lceil n'/3 \rceil - 1$ shares of the output (corresponding to the actively corrupt parties). It follows from Lemma 8 that this information perfectly hides the output which is $(\lceil n'/3 \rceil - 1, 1)$-levelled shared. **(b)** Suppose that Phase 2 is unsuccessful due to failure in reconstruction of summand y_j $(j \geq 2)$. Then it follows from the property of fairness of $\pi_{\mathsf{pLevRec}}^{\lceil n'/3 \rceil - 1,1}$ (Lemma 9) that the adversary cannot learn the summand y_{j-1} and subsequently the output y. Lastly, suppose that there is failure in reconstructing y_1. This will occur only if $t'_a = 0$ and $t'_f = n' - 1$ (as $1 \geq n' - t'_a - t'_f > 2t'_a$ holds when adversary disrupts

Protocol $\pi_{\mathsf{god}}^{t_p=0}$

Inputs: Each party P_i participates with input x_i ($i \in [n]$)
Output: $f(x_1, \ldots, x_n)$
Building Blocks: - A perfectly secure MPC protocol ψ achieving G.O.D against $t < n/3$ *active* corruptions (instantiated by [BTH08]). ψ achieves security with abort in the presence of $\mathcal{A}^{(t_a,t_f)}$ respecting $3t_a + t_f < n$; Function $f_{\mathsf{pLevSh}}^{\alpha,\beta}$ (Fig 9) and Protocol $\pi_{\mathsf{pLevRec}}^{\alpha,\beta}$ (Fig 10)

Initialization: $\mathcal{Q} = \mathcal{P}$, $\mathcal{C} = \emptyset$, $n' = n$, $t'_f = t_f, t'_a = t_a$, $t'_p = t_p = 0$.

Phase 1: Each $P_i \in \mathcal{Q}$ does the following:

- Participate in an execution of ψ with input x_i to compute $f_{\mathsf{pLevSh}}^{\lceil n'/3 \rceil - 1,1}(y)$ where $y = f(x_1, \ldots, x_n)$ (default inputs used for parties in $\mathcal{C}$). [a]
- If a set of parties S with $|S| \geq 1$ crash during ϕ, update $\mathcal{C} = \mathcal{C} \cup S$, $\mathcal{Q} = \mathcal{Q} \setminus S, t'_f = t'_f - |S|$ and $n' = n' - |S|$. Re-run Phase 1.
- Else, proceed to Phase 2.

Phase 2: Let $\langle y \rangle^{\lceil n'/3 \rceil - 1,1}$ denote the output of Phase 1. The parties in $\mathcal{Q}$ do the following:

- Run the reconstruction protocol $\pi_{\mathsf{pLevRec}}^{\lceil n'/3 \rceil - 1,1}(\langle y \rangle^{\lceil n'/3 \rceil - 1,1})$ to reconstruct the $(\lceil n'/3 \rceil - 1, 1)$-levelled shared output y.
- Suppose it outputs $y' \neq \perp$, output y'.
- Else, suppose it outputs $(\perp, S)$. Then, update $\mathcal{C} = \mathcal{C} \cup S$, $\mathcal{Q} = \mathcal{Q} \setminus S, t'_f = t'_f - |S|$ and $n' = n' - |S|$. Restart from Phase 1.

[a] If $n' \leq 3$, ψ is used to compute f directly as there are no active corruptions

Fig. 15. Perfect SFE with G.O.D against dynamic adversary with $t_p = 0$

reconstruction of y_1). In such a case, it follows trivially that the adversary has no information about y as he cannot access any message sent during $\pi_{\mathsf{god}}^{t_p=0}$. This is because all communication throughout $\pi_{\mathsf{god}}^{t_p=0}$ occurs over pairwise-private channel (recall that broadcast in the perfect setting is also realized by adapting standard broadcast protocols which use pairwise-private channels) which an adversary with $t'_a = 0$ will not be able to access. This completes the intuition.

C.3 Reactive MPC with Perfect Security

C.3.1 VSS with G.O.D when $t_a = 0$. Recall the impossibility of perfect VSS where reconstruction is G.O.D. against a dynamic adversary even when $t_f = 0$ (Section 4.2). In this section, we explore the feasibility question for case of $t_a = 0$. In this case the additional condition $2t_p + 2t_f \leq n$ is necessary and sufficient for G.O.D. VSS, as we now show.

Necessity of $2t_p + 2t_f \leq n$. This condition $2t_p + 2t_f \leq n$ can be derived by translating the characterization of general mixed adversaries in [BFH+08], which we elaborate in Lemma 16 of Section 6.2. A simple argument showing the necessity of $2t_p + 2t_f \leq n$ for perfect VSS with G.O.D., even against a weaker dynamic adversary with $t_a = 0$ is as follows - This argument is similar to the proof of Theorem 2 in Section 3.2. Suppose, for contradiction, that there exists a perfect VSS with G.O.D. when $2t_p + 2t_f > n$. Firstly, it must hold that the joint state of any set of parties S (that excludes the dealer), where $|S| < n/2$ must be such that it is identically distributed for all different values of s (where s denotes the secret committed by the dealer). This is dictated by the property of privacy at the end of sharing against an adversary who corrupts the parties in S passively (adversary respecting $2t_p + t_f < n$ chooses $t_p < n/2$). Consider an execution of the VSS protocol where everyone behaves honestly during the sharing phase and the dealer commits to a secret s'. Suppose during reconstruction, the adversary fail corrupts t_f parties including the dealer where $t_f > n/2$ (allowed based on the assumption $2t_p + 2t_f > n$). Then the remaining set of parties S' is such that $|S'| = n - t_f < n/2$. Since the joint state of parties in S' does not have any information about the secret s' committed by the dealer, reconstruction with G.O.D is impossible. This completes the argument of necessity of $2t_p + 2t_f \leq n$ for perfect VSS with G.O.D.

Sufficiency of $2t_p + 2t_f \leq n$. We present a VSS protocol that achieves G.O.D when $2t_p + 2t_f \leq n$. For the sharing protocol, we use our SFE protocol $\pi_{\mathsf{god}}^{t_a=0}$ from Fig. 14 in Section C.2.1 that achieves G.O.D. against a dynamic adversary with $t_a = 0$. This protocol is used to compute $f_{\mathsf{sh}}^{\lceil n/2 \rceil - 1, \mathsf{perf}}(s)$ where the dealer inputs his secret s. The reconstruction protocol involves all parties broadcasting their shares $[s]_{\lceil n/2 \rceil - 1}$ (that was output by the sharing protocol). The parties reconstruct the secret s by simply interpolating a $(\lceil n/2 \rceil - 1)$-degree polynomial using the shares that are broadcast.

The property of privacy (when the dealer is honest) at the end of sharing follows directly from the property of $\lceil n/2 \rceil - 1$-sharing of the secret s, since the

adversary has access only to $t_p \leq \lceil n/2 \rceil - 1$ shares. Correctness follows directly from the correctness of $\pi_{\mathsf{god}}^{t_a=0}$ and the fact that all the shares exchanged during reconstruction are untampered (as no active corruptions are allowed). Lastly, the argument for reconstruction with G.O.D. is as follows: Since there are no active corruptions and $t_f \leq n/2$ (implied by $2t_p + 2t_f \leq n$), the shares broadcast by the honest and passively-corrupt parties (which constitute a set of $n - t_f \geq n/2$ parties) is sufficient to reconstruct the secret robustly.

We state the above result in the following Theorem.

Theorem 14. *The condition $2t_p + 2t_f \leq n$ is necessary and sufficient to design a perfectly-secure VSS protocol that achieves G.O.D against an adversary who can choose (t_p, t_f) such that $2t_p + t_f < n$.*

Since the above VSS and SFE of Section C.2.1 imply reactive MPC (by verifiably secret-sharing the intermediate states [HMZ08]), we obtain the following result.

Theorem 15. *The condition $2t_p + 2t_f \leq n$ is necessary and sufficient to design a perfectly-secure reactive MPC that achieves G.O.D against an adversary who can choose (t_p, t_f) such that $2t_p + t_f < n$.*

C.3.2 VSS with G.O.D when $t_p = 0$. In this section, we analyze the feasibility of perfect VSS with G.O.D against a dynamic adversary with $t_p = 0$ i.e. the adversary must respect $3t_a + t_f < n$. We will show that in this case the additional condition $3t_a + 3/2t_f \leq n$ is necessary and sufficient for G.O.D. VSS.

Necessity of $3t_a + 3/2t_f \leq n$. The proof appears in Lemma 15 of Section 6.2.

Sufficiency of $3t_a + 3/2t_f \leq n$. We present a VSS protocol that achieves G.O.D. when $3t_a + 3/2t_f \leq n$. For the sharing protocol, we use our SFE protocol $\pi_{\mathsf{god}}^{t_p=0}$ from Fig. 15 in Section C.2.2 that achieves G.O.D. against a dynamic adversary with $t_p = 0$. This protocol is used to compute $f_{\mathsf{sh}}^{\delta;\mathsf{perf}}(s)$, where $\delta = \lceil n/3 \rceil - 1$, and the dealer inputs his secret s. The reconstruction protocol involves running $\pi_{\mathsf{PerfRec}}^{\delta}([s]_\delta)$ where $[s]_\delta$ denotes the output of the sharing protocol.

Privacy of the above described VSS protocol (when the dealer is honest) follows from the property of ($\delta = \lceil n/3 \rceil - 1$)-sharing of s (adversary has access only to $t_a \leq \lceil n/3 \rceil - 1$ shares of s). Next, correctness follows from the correctness of $\pi_{\mathsf{god}}^{t_p=0}$ and $\pi_{\mathsf{PerfRec}}^{\delta}([s]_\delta)$. Lastly, the reconstruction is G.O.D due to the following: The number of shares broadcast during $\pi_{\mathsf{PerfRec}}^{\delta}([s]_\delta)$ is at least $|W| = n - t_a - t_f + r$, where $r \leq t_a$ is the number of tampered shares. Recall that $t_a + t_f \leq 2n/3 - t_a$ (as per our assumption $3t_a + 3/2t_f \leq n$), implying that $|W| = n - t_a - t_f + r \geq n/3 + t_a + r > \delta + 2r$ (as $n/3 > \delta$ and $t_a \geq r$). Since $\pi_{\mathsf{PerfRec}}^{\delta}([s]_\delta)$ returns the correct secret when $|W| > \delta + 2r$ (due to property of π_{RSDec}), we can conclude that the reconstruction is G.O.D.

We state the above result in the following Theorem:

Theorem 16. *There exists a perfectly-secure VSS protocol that achieves G.O.D against a dynamic adversary with $t_p = 0$ (who can choose (t_a, t_f) such that $3t_a + t_f < n$) if $3t_a + \frac{3}{2}t_f \leq n$ holds.*

Since the above VSS and SFE of Section C.2.2 imply reactive MPC (by verifiably secret-sharing the intermediate states [HMZ08]), we obtain the following result.

Theorem 17. *There exists a perfectly-secure reactive MPC that achieves G.O.D against a dynamic adversary with $t_p = 0$ (who can choose (t_a, t_f) such that $3t_a + t_f < n$) if $3t_a + \frac{3}{2}t_f \leq n$ holds.*