Beyond Honest Majority: The Round Complexity of Fair and Robust Multi-party Computation

Arpita Patra and Divya Ravi [⋆]

Indian Institute of Science, India
{arpita,divyar}@iisc.ac.in

Abstract. Two of the most sought-after properties of Multi-party Computation (MPC) protocols are fairness and guaranteed output delivery (GOD), the latter also referred to as robustness. Achieving both, however, brings in the necessary requirement of malicious-minority. In a generalised adversarial setting where the adversary is allowed to corrupt both actively and passively, the necessary bound for a n-party fair or robust protocol turns out to be $t_a + t_p < n$, where t_a, t_p denote the threshold for active and passive corruption with the latter subsuming the former. Subsuming the malicious-minority as a boundary special case, this setting, denoted as dynamic corruption, opens up a range of possible corruption scenarios for the adversary. While dynamic corruption includes the entire range of thresholds for (t_a, t_p) starting from $(\lceil \frac{n}{2} \rceil - 1, \lfloor \frac{n}{2} \rfloor)$ to $(0, n-1)$, the boundary corruption restricts the adversary only to the boundary cases of $(\lceil \frac{n}{2} \rceil - 1, \lfloor \frac{n}{2} \rfloor)$ and $(0, n-1)$. Notably, both corruption settings empower an adversary to control majority of the parties, yet ensuring the count on active corruption never goes beyond $\lceil \frac{n}{2} \rceil - 1$.

We target the round complexity of fair and robust MPC tolerating dynamic and boundary adversaries. As it turns out, $\lceil \frac{n}{2} \rceil + 1$ rounds are necessary and sufficient for fair as well as robust MPC tolerating dynamic corruption. The non-constant barrier raised by dynamic corruption can be sailed through for a boundary adversary. The round complexity of 3 and 4 is necessary and sufficient for fair and GOD protocols respectively, with the latter having an exception of allowing 3 round protocols in the presence of a single active corruption. While all our lower bounds assume pair-wise private and broadcast channels and are resilient to the presence of both public (CRS) and private (PKI) setup, our upper bounds are broadcast-only and assume only public setup. The traditional and popular setting of malicious-minority, being restricted compared to both dynamic and boundary setting, requires 3 and 2 rounds in the presence of public and private setup respectively for both fair as well as GOD protocols.

Keywords: Fairness · Guaranteed Output Delivery · MPC · Round Complexity · Dynamic · Boundary

[⋆] This article is a full and extended version of an earlier article to appear in ASIACRYPT 2019.

1 Introduction

Secure multi-party computation (MPC) [GMW87, CDG87, Yao82], which is arguably the most general problem in cryptography, allows a group of mutually distrustful parties to compute a joint function on their inputs without revealing any information beyond the result of the computation. While the distrust amongst the parties is modelled by a centralized adversary $\mathcal{A}$ who can corrupt a subset of the parties, the security of an MPC protocol is captured by a real-world versus ideal-world paradigm. According to this paradigm, adversarial attacks in a real execution of the MPC protocol can be translated to adversarial attacks in the ideal-world where the parties interact directly with a trusted-third party who accepts private inputs, computes the desired function and returns the output to the parties; thereby trivially achieving *correctness* (function output is correctly computed on parties' inputs) and *privacy* ($\mathcal{A}$ learns nothing about the private inputs of honest parties, beyond what is revealed by the output).

Two of the most sought-after properties of MPC protocols are fairness and robustness (alternately, guaranteed output delivery a.k.a. GOD). The former ensures that adversary obtains the output if and only if honest parties do, while the latter guarantees that the adversary cannot prevent honest parties from obtaining the output. Both these properties are trivially attainable in the presence of any number of *passive* (semi-honest) corruption where the corrupt parties follow the protocol specifications but the adversary learns the internal state of the corrupt parties. However, in the face of stringent *active* (malicious) corruption where the parties controlled by the adversary deviate arbitrarily from the protocol; fairness and GOD can be achieved only if the adversary corrupts at most a minority of the parties (referred to as malicious minority) [Cle86].

Opening up the possibility of corrupting parties in both passive and active style, the generalized feasibility condition for a n-party fair or robust protocol turns out to be $t_a + t_p < n$, where t_a, t_p denote the threshold for active and passive corruption, with the latter subsuming the former [HLM13]. We emphasize that t_p is a measure of the *total* number of passive corruptions that includes the actively corrupt parties; therefore the feasibility condition $t_a + t_p < n$ implies $t_a \leq \lceil n/2 \rceil - 1$. In its most intense and diverse avatar, referred as *dynamic-admissible*, the adversary can take control of the parties in one of the ways drawn from the entire range of admissible possibilities of (t_a, t_p) starting from $(\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ to $(0, n-1)$[1]. In a milder setting, referred as *boundary-admissible*, the adversary is restricted only to the boundary cases, namely $(\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ and $(0, n-1)$. Subsuming the traditional malicious-minority and passive-majority (majority of the parties controlled by passive adversary) setting for achieving fairness and GOD as special cases, both dynamic as well as boundary setting give the adversary more freedom and consequently more strength to the protocols.

[1] We refer to the dynamic-admissible adversary as dynamic adversary, which is not to be confused with the notion of adaptive adversaries who are allowed to dynamically choose which parties to corrupt during the protocol execution.

Notably, both empower an adversary to control majority of the parties, yet ensuring the count on active corruption never goes beyond $\lceil \frac{n}{2} \rceil - 1$.

The study of protocols in dynamic and boundary setting is well motivated and driven by theoretical and practical reasons. Theoretically, the study of generalized adversarial corruptions gives deeper insight into how passive and active strategies combine to influence complexity parameters of MPC such as efficiency, security notion achieved and round complexity. Practically, the protocols in dynamic and boundary setting offer strong defence and are more tolerant and better-fit in practical scenarios where the attack can come in many unforeseen ways. Indeed, deploying such protocols in practice is far more safe than traditional malicious-minority and passive-majority protocols that completely break down in the face of boundary adversaries, let alone dynamic adversaries. For instance, consider MPC in server-aided setting where instead of assuming only actively corrupt clients and honest servers, the collusion of client-server is permitted where some of the servers can be passively monitored. This model is quite realistic as it does not contradict the reputation of the system (since the passive servers follow protocol specifications and can thereby never be exposed / caught). The option of allowing corruption in both passive and active styles is quite relevant in such scenarios. Driven by the above credible reasons and extending the study of exact round complexity of fair and robust protocols beyond the traditional malicious-minority setting [GIKR02, GLS15, PR18a], in this work, we aim to settle the same for the regime of dynamic and boundary corruption.

Related Work. We begin with outlining the most relevant literature of round complexity of fair and robust MPC protocols in the traditional adversarial settings involving only single type of adversary (either passive or active). To begin with, 2 rounds are known to be necessary to realize any MPC protocol, regardless of the type of adversary, no matter whether a setup is assumed or not as long as the setup (when assumed) is independent of the inputs of the involved parties [HLP11]. A 1-round protocol is susceptible to "residual function attack" where an adversary can evaluate the function on multiple inputs by running the computation with different values for his inputs with fixed inputs for the honest parties. The result of [GIKR02] shows necessity of 3 rounds for fairness in the plain and CRS setting (assumes parties have access to a common reference string), when the number of malicious corruptions is at least 2 (i.e. $t \geq 2$), irrespective of the number of parties, assuming the parties are connected by pairwise-private and broadcast channels. Complementing this result, the lower bound of [PR18a] extends the necessity of 3 rounds for any t (including $t = 1$) as long as $n/3 \leq t < n/2$. The work of [GLS15] shows 3 to be the lower bound for fairness in the presence of CRS, assuming broadcast-only channels (no private channels).

In terms of the upper bounds, the works of [GS18, BL18] showed that 2-rounds are sufficient to achieve robustness in the passive-majority setting. In accordance with the impossibility of [Cle86] and sufficiency of honest-majority shown by classical result of [RB89], the upper bounds in the malicious setting involve $t < n/2$ parties. These include the 3-round constructions of [GLS15, ACGJ18, BJMS18] based on tools such as Zaps, multi-key FHE, dense cryptosystems. The protocol of [GLS15] can be collapsed to two rounds given access to a PKI (public-key infrastructure). In the information-theoretic setting involving $t < n/4$ malicious corruptions, the work of [ABT19] presents a 3-round perfectly-secure robust protocol. In the domain of small-number of parties, round optimal protocols achieving fairness and robustness appear in [IKKP15, PR18a].

Moving on to the setting of generalized adversary, there are primarily two adversarial models that are most relevant to us. The first model initiated by [DDWY93] consider a mixed adversary (referred to as graceful degradation of *corruptions*) that can *simultaneously* perform different types of corruptions. Feasibility results in this model appeared in the works of [FHM98, FHM99, HMZ08, BFH$^+$08]. The dynamic-admissible adversary considered in our work is consistent with this model since it involves simultaneous active and passive corruptions. The second model proposed by [Cha89] concerns protocols that are secure against an adversary that can either choose to corrupt a subset of parties with particular corruption type (say, passively) or alternately a different subset (typically smaller) of parties with a second corruption type (say, actively), but only *single* type of corruption occurs at a time. Referred to as graceful degradation of *security* [Cha89, LRM10, FHHW03, FHW04, IKLP06, Kat07, IKK$^+$11], such protocols achieve different security guarantees based on the set of corrupted parties; for instance robustness/information-theoretic security against the smaller corruption set and abort/computational security against the larger corruption set. We note that the boundary-admissible adversary when n is odd, involves either purely active (since $t_a = t_p$ holds when $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor))$ corruptions or purely passive corruptions (where $(t_a, t_p) = (0, n-1)$); thereby fitting in the second model (In fact, boundary-admissible adversary for odd n degenerates to the adversarial model studied in "best-of-both-worlds" MPC [IKK$^+$11]). However, in case of even n, the boundary-admissible adversary with $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ would involve simultaneous passive and active corruption as $t_p = t_a + 1$ and fit in the prior model. Lastly, both graceful degradation of security and corruptions were generalized in the works of [HLMR11, HLM13]. To the best of our knowledge, the interesting and natural question of round complexity has not been studied in these stronger adversarial models.

1.1 Our Results

In this work, we target and resolve the exact round complexity of fair and robust MPC protocols in both dynamic and boundary setting. This is achieved via 3 lower bounds that hold assuming *both* CRS and PKI setup and 5 upper bounds that assumes CRS *alone*. Notably, the lower bounds in PKI (private) setup extend to a model with arbitrary correlated randomness. In terms of network setting, while our lower bounds hold assuming *both* pairwise-private and broadcast channels, all our upper bounds use broadcast channel *alone*. All our upper bounds are generic compilers that transform a 2-round protocol achieving unanimous abort (either all honest parties obtain output or none of them do) or identifiable abort (corrupt parties are identified in case honest parties do not obtain the output) against malicious majority to a protocol achieving the stronger guarantees of fairness/robustness against stronger adversaries (namely, dynamic and boundary adversaries). The need for CRS in our constructions stems from the underlying 2-round protocol achieving unanimous or identifiable abort. We leave open the question of constructing tight upper bounds or coming up with new lower bounds in the plain model. We elaborate on the results below.

Dynamic Adversary. We recall that in this challenging setting, the adversary has the freedom to choose from the entire range of thresholds for (t_a, t_p) starting from $(\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ to $(0, n-1)$. Our first lower bound establishes that $\lceil n/2 \rceil + 1$ rounds are necessary to achieve fairness against dynamic adversary. Since robustness is a stronger security notion, the same lower bound holds for GOD as well. This result not only rules out the possibility of constant-round fair protocols but also gives the *exact* lower bound. We give two matching upper bounds, one for fairness and the other for robustness, where the former is subsumed by and acts as a stepping stone to the latter. These results completely settle the round complexity of this setting in the CRS model.

Boundary Adversary. The leap in round complexity ebb in the milder boundary adversarial setting where adversary is restricted to the boundary cases of $(\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ and $(0, n-1)$. Our two lower bounds of this setting show that 4 and 3 rounds are necessary to achieve robustness and fairness respectively against the boundary adversary. Our first 4-round lower bound is particularly interesting, primarily due to two reasons. (1) As mentioned earlier, when n is odd, the boundary cases reduce to pure active ($t_a = t_p$ when $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$) and pure passive ($(t_a, t_p) = (0, n-1)$) corruptions. We note that security against malicious-minority and passive-majority are known to be attainable independently in just 2 rounds assuming access to CRS and PKI [GLS15, GS18, BL18]. Hence, our 4-round lower bound encapsulates the difficulty in designing protocols tolerant against an adversary who can choose among his two boundary corruption types arbitrarily. (2) This lower bound can be circumvented in case of single malicious corruption i.e. against a special-case boundary adversary restricted to corruption scenarios $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ and $(t_a, t_p) = (0, n-1)$. (We refer to such an adversary as special-case boundary adversary with $t_a \leq 1$). This observation augments the rich evidence in literature [PCRR09, BKP11, IKKP15] which show the impact of single corruption on feasibility results. With respect to our second lower bound for fairness against boundary adversary, we first note that the 3-round lower bound for fairness in the presence of CRS is trivial given the feasibility results of [GIKR02, GLS15, PR18a]. However, they break down assuming access to PKI. Thus, the contribution of our second lower bound is to show that the 3-round lower bound holds for boundary adversary even in the presence of PKI.

We complement these two lower bounds by three tight upper bounds. The upper bounds achieving robustness include a 4-round protocol for the general case and a 3-round protocol for the special-case of one malicious corruption that demonstrates the circumvention of our first lower bound. Lastly, our third upper bound is a 3-round construction achieving fairness, demonstrating the tightness of our second lower bound.

Our results appear in the table below with comparison to the round complexity in the traditional settings of achieving fairness and robustness. Since PKI (private) setup subsumes CRS (public) setup which further subsumes plain model (no setup), the lower and upper bounds are specified with their maximum tolerance and minimum need respectively amongst these setup assumptions. The results provide us further insights regarding how disparity in adversarial setting affects round complexity. Note that the round complexity of fair protocols in the CRS model against an adversary corrupting minority of parties maliciously, remains unaffected in the setting of boundary adversary; which is a stronger variant of the former. On the other hand, this switch of adversarial setting causes the lower bound of robust protocols in the model assuming both CRS and PKI to jump from 2 to 4. Lastly, the gravity of dynamic corruption on round complexity is evident in the leap from constant-rounds of $3, 4$ in the boundary corruption case to $\lceil n/2 \rceil + 1$.

Adversary	Security	Rounds	Lower bound	Upper Bound
Passive-majority	Fair, GOD	2	[HLP11] (private)	[GS18, BL18] (plain)
Malicious-minority	Fair, GOD	3	[GLS15, PR18a] (public)	[ACGJ18, BJMS18] (plain)
	Fair, GOD	2	[HLP11] (private)	[GLS15] (private)
Boundary	Fair	**3**	**[This, Thm. 5]** (private)	**[This, Thm. 8]** (public)
	GOD	**4** (**3** when $t_a \leq 1$)	**[This, Thm. 4, 5]** (private)	**[This, Thm. 6,7]** (public)
Dynamic	Fair, GOD	$\lceil \frac{\mathbf{n}}{\mathbf{2}} \rceil + \mathbf{1}$	**[This, Thm. 1]** (private)	**[This, Thm. 2, 3]** (public)

1.2 Techniques

In this section, we give a glimpse into the techniques used in our lower bounds and matching upper bound constructions.

Lower Bounds. We present 3 lower bounds, all of which hold assuming access to *both* CRS and PKI– **(a)** $\lceil n/2 \rceil + 1$ rounds are necessary to achieve fairness against dynamic adversary. **(b)** 4 rounds are necessary to achieve robustness against a boundary adversary. **(c)** 3 rounds are necessary to achieve fairness against a boundary adversary.

The first lower bound **(a)** effectively captures the power of dynamic corruption stemming from the ambiguity caused by the total range of thresholds (t_a, t_p) starting from $(\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ to $(0, n-1)$. The proof navigates through this sequence starting with maximal active corruption and proceeds to scenarios of lesser active corruptions one at a time. An inductive argument neatly captures how the value of t_p growing alongside decreasing values of t_a can be exploited by adversarial strategies violating fairness, eventually dragging the round complexity all the way upto $\lceil n/2 \rceil + 1$. The lower bounds **(b)** and **(c)** are shown by considering a specific set of small number of parties and assume the existence of a 3 (2) round robust (fair) protocol for contradiction respectively. Subsequently, inferences are drawn based on cleverly-designed strategies exploiting the properties of GOD and fairness. These inferences and strategies are interconnected in a manner that builds up to a strategy violating privacy, thereby leading to a final contradiction.

Upper Bounds. We present 5 upper bounds, in the broadcast-only setting comprising of two upper bounds each for fairness and GOD against dynamic and boundary adversary respectively and lastly, an additional 3-round upper bound for GOD against the special case of single malicious corruption by boundary adversary in order to demonstrate the circumvention of lower bound **(b)**. Tightness of this upper bound follows from lower bound **(c)** (that holds for single malicious corruption) as GOD implies fairness. Our upper bounds can be viewed as “compiled" protocols obtained upon plugging in any 2-round broadcast-only protocols [GS18, BL18] achieving unanimous abort against malicious majority. While the fair upper-bounds do not require any additional property from the underlying 2-round protocol, our robust protocols demand the property of *identifiable abort* and *function-delayed* property i.e. the first round of the protocol is independent of the function to be computed and the number of parties. Looking ahead, this enables us to run many parallel instances of the round 1 in the beginning and run the second round sequentially as and when there is a failure, to compute a new function (that gets determined based on the identities of the corrupt parties). Assumption wise, all our upper bound constructions rely on 2-round maliciously-secure oblivious transfer (OT) in common random/reference string models. We now give a high-level overview of the specific challenges we encounter in each of our upper bounds and the techniques we use to tackle them.

Dynamic adversary: The two upper bounds against dynamic adversary show sufficiency of $\lceil n/2 \rceil + 1$ rounds to achieve fairness and robustness against dynamic admissible adversary. The upper bound for fairness is built upon the protocol of [HLM13] that introduces a special-kind of sharing, which we refer to as levelled-sharing where a value is divided into summands (adding upto the value) and each summand is shared with varying degrees. The heart of the protocol of [HLM13] lies in its gradual reconstruction of the levelled-shared output (obtained by running an MPC protocol with unanimous abort), starting with the summand corresponding to the highest degree down to the lowest. The argument for fairness banks on the fact that the more the adversary raises its disruptive power in an attempt to control reconstruction of more number of summands, the more it looses its eavesdropping capability and consequently learns fewer number of summands by itself and vice versa. This discourages an adversary from misbehaving as using maximal disruptive power reduces its eavesdropping capability such that he falls short of learning the next summand in sequence without the help of honest parties. The innovation of our fair protocol lies in delicately fixing the parameters of levelled-sharing in a manner that optimal round complexity can be attained whilst maintaining fairness.

Next, we point that since the fair protocol consumes the optimal round complexity of $\lceil n/2 \rceil + 1$ even in the case of honest execution, the primary hurdle in our second upper bound is to be able to carry out re-runs when an adversary disrupts computation to achieve robustness without consuming extra rounds. Banking on the player-elimination technique, we use identifiability to bar the corrupt parties disrupting computation from participating thereafter. Having parallel execution of Round 1 of all the required re-reruns helps us get closer to the optimal bound. While these approaches aid to a great extent, the final saviour comes in the form of a delicate and crucial observation regarding how the thresholds of the levelled-sharing can be manipulated carefully, accounting for the cheaters identified so far. This trick exploits the pattern of reduced corruption scenarios obtained upon cheater identification and helps to compensate for the rounds consumed in subprotocols that were eventually disrupted by the adversary. The analysis of the round complexity of the protocol being subtle, we use an intricate recursive argument to capture all scenarios and show that the optimal lower bound is never exceeded. Lastly, we point that both upper bound constructions against dynamic adversary assume equivocal non-interactive commitment (such as Pedersen commitment [Ped91]).

Boundary adversary: The three upper bounds against boundary-admissible adversary restricted to corruption scenarios either $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$ show that **(a)** 4 rounds are sufficient to achieve robustness against boundary-admissible adversary **(b)** 3 rounds are sufficient to achieve robustness against special-case boundary-admissible adversary when $t_a \leq 1$ i.e. adversary corrupts with parameters either $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$ **(c)** 3 rounds are sufficient to achieve fairness against boundary-admissible adversary.

At a high-level, all the three upper bounds begin with a 2-round protocol secure against malicious majority that computes threshold sharing of the output. Intuitively, this seems to serve as the only available option as protocols customized for malicious minority typically breach privacy when views of majority of the parties are combined (thereby will break down against $t_p < n$ semi-honest corruptions). On the flip side, protocols customized for exclusively passive majority may violate correctness/privacy in the presence of even single malicious corruption. Subsequently, this natural route bifurcates into two scenarios based on whether the adversary allows the computation of the threshold sharing of output to succeed or not. In case of success, all the three upper bounds proceed via the common route of reconstruction which is guaranteed to be robust by the property of threshold sharing. The distinctness of the 3 settings (accordingly the upper bounds) crops up in the alternate scenario i.e. when the computation of threshold sharing of output aborts. While in upper bound **(c)**, parties simply terminate with $\perp$ maintaining fairness enabled by privacy of the threshold sharing; the upper bounds **(a)** and **(b)** demanding stronger guarantee of robustness cannot afford to do so. These two upper bounds exploit the fact that the corruption scenario has now been identified to be the boundary case having active corruptions, thereby protocols tolerating malicious minority can now be executed.

While the above outline is inspired by the work of [IKK$^+$11], we point that we need to tackle the exact corruption scenarios as that of the protocols of [IKK$^+$11] only when n is odd. On the other hand when n is even, the extreme case for active corruption accommodates an additional passive corruption ($t_p = t_a + 1$). Apart from hitting the optimal round complexity, tackling the distinct boundary cases for odd and even n in a unified way brings challenge for our protocol. To overcome these challenges, in addition to techniques of identification and elimination of corrupt parties who disrupt computation, we employ tricks such as parallelizing without compromising on security to achieve the optimum round complexity. Lastly, we point that the upper bound **(a)** assumes Zaps (2-round, public-coin witness-indistinguishable protocols) and public-key encryption.

2 Preliminaries

We consider a set of parties $\mathcal{P} = \{P_1, \ldots, P_n\}$. Our upper bounds assume the parties communicate over a broadcast channel and a setup where parties have access to a common reference string (CRS). Our lower bounds hold even when the parties are additionally connected by pairwise-secure and authentic channels and for a stronger setup, namely assuming access to CRS as well as public-key infrastructure (PKI). Each party is modelled as a probabilistic polynomial-time (PPT) Turing machine. We assume that there exists a PPT adversary $\mathcal{A}$, who can corrupt a subset of these parties.

We consider two kinds of adversarial settings in this work. In both settings, the adversary $\mathcal{A}$ is characterised by two thresholds (t_a, t_p), where he may corrupt upto t_p parties passively, and upto t_a of these parties even actively. Note that t_p is the total number of passive corruptions that includes the active corruptions and additional parties that are exclusively passively corrupt. We now define dynamic and boundary admissible adversaries.

Definition 1 (Dynamic-admissible Adversary). *An adversary attacking an n-party MPC protocol with threshold (t_a, t_p) is called dynamic-admissible as long as $t_a + t_p < n$ and $t_a \leq t_p$.*

Definition 2 (Boundary-admissible Adversary). *An adversary attacking an n-party MPC protocol with threshold (t_a, t_p) is called boundary-admissible as long as he corrupts either with parameters (a) $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ or (b) $(t_a, t_p) = (0, n-1)$.*

In our work, we also consider a special-case of boundary adversary with $t_a \leq 1$ where the adversary corrupts either with parameters $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$.

Notation. We denote the cryptographic security parameter by κ. A negligible function in κ is denoted by $\mathtt{negl}(\kappa)$. A function $\mathtt{negl}(\cdot)$ is negligible if for every polynomial $p(\cdot)$ there exists a value N such that for all $m > N$ it holds that $\mathtt{negl}(m) < \frac{1}{p(m)}$. Composition of two functions, f and g (say, $h(x) = g(f(x))$)

is denoted as $g \diamond f$. We use $[n]$ to denote the set $\{1, \ldots n\}$ and $[a, b]$ to denote the set $\{a, a+1 \ldots b\}$ when $a \leq b$ or the set $\{a, a-1, \ldots b\}$ when $a > b$. Lastly, for dynamic-admissible adversary, we denote the set of actively and passively corrupt parties by $\mathcal{D}$ and $\mathcal{E}$ respectively, where $|\mathcal{D}| = t_a$ and $|\mathcal{E}| = t_p$.

Security definition and the functionalities. The security definition (based on the standard real/ideal world paradigm) and the functionalities appear in Appendix A. Since we consider deterministic functionalities, the security guarantees of correctness and privacy are analyzed separately [Lin17] in all our security proofs.

Road map. Our lower and upper bounds for dynamic and boundary corruption appear in Sections 3-4 and in Sections 5-6 respectively.

3 Lower Bounds for Dynamic Corruption

In this section, we show that $\lceil \frac{n}{2} \rceil + 1$ rounds are necessary to achieve MPC with fairness against a dynamic-admissible $\mathcal{A}$ with threshold (t_a, t_p). This result shows impossibility of constant-round fair and robust protocols against dynamic adversary, assuming access to CRS and PKI. Notably, this lower bound extends to a model with arbitrary correlated randomness.

We begin with a high-level description of the proof. Towards a contradiction, we assume that there exists a $\lceil \frac{n}{2} \rceil$-round n-party MPC protocol π computing any function f that achieves fairness against a dynamic-admissible $\mathcal{A}$ in the presence of a setup with CRS and PKI. Next, we define a sequence of hybrids of π, that navigate through all the possible admissible corruption scenarios assuming $t_a + t_p = n - 1$ and starting with the maximum admissible value of $t_a = \lceil n/2 \rceil - 1$.

Our first hybrid under the spell of a dynamic-admissible adversary, corrupting $\lceil n/2 \rceil - 1$ parties actively and stopping their communication in the last round, lets us conclude that the joint view of the honest and passively-corrupted parties by the end of penultimate round must hold the output in order for π to satisfy fairness. If not, while ceasing communication in the last round does not prevent $\mathcal{A}$ from getting all the messages in the last round and thereby the output, the honest parties do fail to compute the output due to the non-cooperation of t_a parties, violating fairness. The views of the passively corrupt parties need to be taken into account as they follow protocol steps correctly and assist in output computation.

Leveraging the fact that drop of t_a leads to rise of t_p, we then propose a new hybrid where t_a is demoted by 1 and consequently t_p grows big enough to subsume the list of honest and passive-corruption from the previous hybrid. As the view of the adversary in this hybrid holds the output by the end of penultimate round itself, its actively-corrupt parties need not speak in the penultimate round. Now fairness in the face of current strategy of the actively-corrupted parties needs the joint view of the honest and passively-corrupted parties by the end of $\lceil n/2 \rceil - 2$ round to hold the output. This continues with the set of honest and passively-corrupted parties growing by size one between every two hybrids.

Propagating this pattern to the earlier rounds eventually lets us conclude that an adversary with threshold $(t_a, t_p) = (0, n-1)$ (no active corruption case) can obtain the output at the end of Round 1 itself. This leads us to a final strategy that violates privacy of π via residual attack. This completes the proof sketch which we formalize below.

Theorem 1. *No $\lceil \frac{n}{2} \rceil$-round n-party MPC protocol can achieve fairness tolerating a dynamic-admissible adversary $\mathcal{A}$ with threshold (t_a, t_p) in a setting with pairwise-private and broadcast channels, and a setup that includes* CRS *and* PKI.

Proof. We prove the theorem by contradiction. Suppose there exists a $\lceil \frac{n}{2} \rceil$-round n-party MPC protocol π computing any function $f(x_1, \ldots, x_n)$ (where x_i denotes the input of party P_i) that achieves fairness against a dynamic-admissible adversary $\mathcal{A}$ with corruption threshold (t_a, t_p) and in the presence of a setup with CRS and PKI.

Consider an execution of π where x_i denotes the input of P_i. We analyze a sequence of hybrids. In each hybrid, the adversary uses honest inputs for corrupted parties who execute the protocol honestly but may abort i.e. drop messages after a particular round in the protocol. Following is the description of the hybrids, where $\mathcal{D}^\ell$ ($\mathcal{E}^\ell$) denotes the set of actively (passively) corrupt parties in hybrid $\textsc{hyb}_\ell$ (where $\mathcal{D}^\ell \subseteq \mathcal{E}^\ell$).

$\textsc{hyb}_1$: $\mathcal{A}$ chooses corruption threshold $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ and does the following. The set of actively corrupt parties $\mathcal{D}^1$ behave honestly upto (and including) Round $\lceil \frac{n}{2} \rceil - 1$ and simply remain silent in the last round i.e. the $\lceil \frac{n}{2} \rceil^{\text{th}}$ round.

$\textsc{hyb}_2$: $\mathcal{A}$ chooses corruption threshold $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - 2, \lfloor n/2 \rfloor + 1)$ where $\mathcal{E}^2 = \mathcal{P} \setminus \mathcal{D}^1$ and does the following. The set of actively corrupt parties $\mathcal{D}^2$ behave honestly upto (and including) Round $\lceil \frac{n}{2} \rceil - 2$ and simply remain silent from Round $(\lceil \frac{n}{2} \rceil - 1)$ onwards.

We generalize the above description to define the remaining sequence i.e. $\textsc{hyb}_3$ to $\textsc{hyb}_{\lceil n/2 \rceil}$.

$\textsc{hyb}_\ell$: $\mathcal{A}$ chooses corruption threshold $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - \ell, \lfloor n/2 \rfloor + \ell - 1)$ where $\mathcal{E}^\ell = \mathcal{P} \setminus \mathcal{D}^{\ell-1}$ and does the following. The set of actively corrupt parties $\mathcal{D}^\ell$ behave honestly upto (and including) Round $\lceil \frac{n}{2} \rceil - \ell$ and simply remain silent from Round $(\lceil \frac{n}{2} \rceil - \ell + 1)$ onwards.

We present a sequence of lemmas to complete the proof. Let $\mathsf{p} = 1 - \mathtt{negl}(\kappa)$ denote the overwhelming probability with which security of π holds, where the probability is defined over the choice of setup and the random coins used by the parties.

Lemma 1. *In $\textsc{hyb}_1$, $\mathcal{A}$ obtains the output $y = f(x_1, x_2, \ldots, x_n)$ with probability at least* p.

Proof. Consider $\textsc{hyb}_1$. Since $\mathcal{A}$ receives all the desired communication throughout the protocol, its view is identically distributed to the view of an honest party in an execution where everyone behaves honestly. Hence, it follows directly from correctness of π (which must hold with probability at least p) that $\mathcal{A}$ must be able to compute the output, with probability at least p. □

Lemma 2. *Suppose $\mathcal{A}$ obtains the output $y = f(x_1, x_2, \ldots, x_n)$ with probability at least q in $\textsc{hyb}_{\ell-1}$, where $\ell \in [2, \lceil n/2 \rceil]$. Then, in $\textsc{hyb}_\ell$, $\mathcal{A}$ must obtain y at the end of Round $(\lceil \frac{n}{2} \rceil - \ell + 1)$, with probability at least $\mathsf{q} \times \mathsf{p}$.*

Proof. Firstly, we analyze $\textsc{hyb}_{\ell-1}$. Fairness of π dictates that whenever $\mathcal{A}$ obtains the output in $\textsc{hyb}_{\ell-1}$ (which occurs with probability at least q), the honest parties must also be able to compute the output even though parties in $\mathcal{D}^{\ell-1}$ stopped communicating from Round $(\lceil \frac{n}{2} \rceil - \ell + 2)$ onwards. Since fairness holds with probability at least p, the combined view of parties in $\mathcal{P} \setminus \mathcal{D}^{\ell-1}$ at the end of Round $(\lceil \frac{n}{2} \rceil - \ell + 1)$ itself must suffice to compute the output with probability at least $\mathsf{q} \times \mathsf{p}$. Next, we note that the view of $\mathcal{A}$ in $\textsc{hyb}_\ell$ is identically distributed to the combined view of parties in $\mathcal{P} \setminus \mathcal{D}^{\ell-1}$ in $\textsc{hyb}_{\ell-1}$. We can thus conclude that in $\textsc{hyb}_\ell$, $\mathcal{A}$ can compute the output at the end of Round $(\lceil \frac{n}{2} \rceil - \ell + 1)$ with probability at least $\mathsf{q} \times \mathsf{p}$.

□

Lemma 3. *In $\textsc{hyb}_\ell$ ($\ell = 1$ to $\lceil n/2 \rceil$), $\mathcal{A}$ obtains the output $y = f(x_1, x_2, \ldots, x_n)$ at the end of Round $(\lceil \frac{n}{2} \rceil - \ell + 1)$, with probability p^ℓ.*

Proof. The proof follows from Lemma 1 and Lemma 2. □

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

Proof. Let $\delta = \lceil n/2 \rceil$. It follows from Lemma 3 that when $\textsc{hyb}_\delta$ occurs, $\mathcal{A}$ obtains the output y at the end of Round 1 itself, with probability $\mathsf{p}^\delta = (1 - \mathtt{negl}(\kappa))^\delta \geq 1 - \delta \times \mathtt{negl}(\kappa) = 1 - \lceil n/2 \rceil \times \mathtt{negl}(\kappa)$ (using binomial expansion $(1-x)^\delta \geq 1 - \delta x$ when $0 < \delta x < 1$) which is overwhelming.

Thus, $\mathcal{A}$ corrupting a set of $t_p = n - 1$ parties passively in $\textsc{hyb}_\delta$, say $\mathcal{E}^\delta = \{P_1, \ldots, P_{n-1}\}$, can execute the residual attack as follows. Compute multiple evaluations of the function f by locally plugging in different values for $\{x_1, \ldots, x_{n-1}\}$ while honest P_n's input x_n remains fixed. This residual function attack which can be executed with overwhelming probability, violates privacy of P_n. As a concrete example, let f be a common output function computing $x_1 \wedge x_n$, where x_i $(i \in \{1, n\})$ denotes a single bit. During the execution of π, $\mathcal{A}$ behaves honestly with input $x_1 = 0$ on behalf of P_1. However, the passively-corrupt P_1 can locally plug-in $x_1 = 1$ and learn x_n (via the output $x_1 \wedge x_n$) with overwhelming probability. This is a clear breach of privacy, as in the ideal world, $\mathcal{A}$ participating honestly with input $x_1 = 0$ on behalf of P_1 would learn nothing about x_n; in contrast to the execution of π where $\mathcal{A}$ learns x_n with overwhelming probability regardless of his input. This completes the proof. □

We have thus arrived at a contradiction to our assumption that π securely computes f and achieves fairness. This completes the proof of Theorem 1.

□

For better understanding, we illustrate the adversarial strategies and implications derived with respect to the specific case of $n = 7$ and 4-round ($\lceil n/2 \rceil = 4$) protocol π in the Table below.

Table 1: Illustration of the lower bound with respect to $n = 7$ and 4-round protocol π. The last column (S, r) indicates the implication that the combined view of parties in S $(= \mathcal{P} \setminus \mathcal{D})$ at the end of Round number r suffices to compute the output with overwhelming probability.

(t_a, t_p)	$\mathcal{D}$	$\mathcal{E}$	Strategy of $\mathcal{A}$	S, r
$(3, 3)$	$\mathcal{D}^1 = \{P_1, P_2, P_3\}$	$\mathcal{E}^1 = \{P_1, P_2, P_3\}$	Stop $\mathcal{D}^1$ after R3	$\{P_4, P_5, P_6, P_7\}$, R3
$(2, 4)$	$\mathcal{D}^2 = \{P_6, P_7\}$	$\mathcal{E}^2 = \{P_4, P_5, P_6, P_7\}$ (i.e. $\mathcal{P} \setminus \mathcal{D}^1$)	Stop $\mathcal{D}^2$ after R2	$\{P_1, P_2, P_3, P_4, P_5\}$, R2
$(1, 5)$	$\mathcal{D}^3 = \{P_1\}$	$\mathcal{E}^3 = \{P_1, P_2, P_3, P_4, P_5\}$ (i.e. $\mathcal{P} \setminus \mathcal{D}^2$)	Stop $\mathcal{D}^3$ after R1	$\{P_2, P_3, P_4, P_5, P_6, P_7\}$, R1
$(0, 6)$	$\mathcal{D}^4 = \emptyset$	$\mathcal{E}^4 = \{P_2, P_3, P_4, P_5, P_6, P_7\}$ (i.e. $\mathcal{P} \setminus \mathcal{D}^3$)	Residual attack	$- - - -$

4 Upper bounds for Dynamic Corruption

In this section, we describe two n-party upper bounds tolerating a dynamic-admissible adversary $\mathcal{A}$ with threshold (t_a, t_p). The first upper bound achieves fairness and is a stepping stone to the construction of the second upper bound that achieves guaranteed output delivery. Both the upper bounds comprise of $\lceil n/2 \rceil + 1$ rounds in the presence of CRS, tightly matching our lower bound result of Section 3. We start with an important building block needed for both the fair and GOD protocols.

4.1 Levelled-sharing of a secret

Our protocols in the dynamic corruption setting involve a special kind of sharing referred as levelled sharing, which is inspired by and a generalized variant of the sharing defined in [HLM13]. The sharing is parameterized with two thresholds, α and β with $\alpha \geq \beta$, that dictate the number of levels as $\alpha - \beta + 1$. To share a secret in (α, β)-levelled-shared fashion, $\alpha - \beta + 1$ additive shares (levels) of the secret, indexed from α to β are created and each additive share is then Shamir-shared [Sha79] using polynomial of degree that is same as its assigned index. Further each Shamir-sharing is authenticated using a non-interactive commitment scheme, to ensure detectably of correct reconstruction. For technical reasons in the simulation-based security proof, we need an instantiation of commitment scheme that allows equivocation of commitment to any message with the help of trapdoor and provides statistical hiding and computational binding. Denoting such a commitment scheme by eNICOM (Equivocal Non-Interactive Commitment), we present both the formal definition and an instantiation based on Pedersen's commitment scheme [Ped91] in Appendix A.1. While the sharing will involve the entire population $\mathcal{P}$ in our fair protocol, it may be restricted to many different subsets of $\mathcal{P}$, each time after curtailing identified actively corrupt parties. The definition therefore is formalized with respect to a set $\mathcal{Q} \subseteq \mathcal{P}$.

Definition 3 ((α, β)-levelled sharing). *A value v is said to be (α, β)-levelled-shared with $\alpha \geq \beta$ amongst a set of parties $\mathcal{Q} \subseteq \mathcal{P}$ if every honest or passively corrupt party P_i in $\mathcal{Q}$ holds L_i as produced by $f_{\mathsf{LSh}}^{\alpha,\beta}(v)$ given in Fig.1.*

Function $f_{\mathsf{LSh}}^{\alpha,\beta}(v)$

1. Choose uniformly random summands $s_\alpha, s_{\alpha-1}, \ldots s_\beta$ with $\sum_{i=\beta}^{\alpha} s_j = v$
2. For $j \in [\alpha, \beta]$, do the following:
   - Choose a random polynomial $g_j(x)$ of degree j with $g_j(0) = s_j$.
   - Sample the public parameter for eNICOM (Section A.1) as $(\mathsf{epp}, t) \leftarrow \mathsf{eGen}(1^\kappa)$. For each share $s_{jk} = g_j(k)$, run $(c_{jk}, o_{jk}) \leftarrow \mathsf{eCom}(\mathsf{epp}, s_{jk}; r_{jk})$ $(P_k \in \mathcal{Q})$ where r_{jk} denotes randomness.
3. Set $L_i = \left(\{s_{ji}, o_{ji}\}_{j \in [\alpha,\beta]}, \{c_{jk}\}_{j \in [\alpha,\beta], P_k \in \mathcal{Q}}\right)$ for $P_i \in \mathcal{Q}$.

Fig. 1: Function $f_{\mathsf{LSh}}^{\alpha,\beta}$ for computing (α, β)-levelled sharing

In our protocols the function $f_{\mathsf{LSh}}^{\alpha,\beta}$ will be realized via an MPC protocol, whereas, given the (α, β)-levelled-sharing, we will use a levelled-reconstruction protocol $\mathsf{LRec}^{\alpha,\beta}$ that enforces reconstruction of the summands one at a time starting with s_α. This levelled reconstruction ensures a remarkable property tolerating any dynamic-admissible adversary– if the adversary can disrupt reconstruction of s_i, then it cannot learn s_{i-1} using its eavesdropping power. This property is instrumental in achieving fairness against the strong dynamic-admissible adversary. The protocol is presented in Fig. 2. Its properties and round complexity are stated below. Note that starting with the feasibility condition $t_a + t_p < n = |\mathcal{P}|$, expelling a set of actively corrupt parties, say $\mathcal{B}$, makes the following impact on t_a, t_p and $\mathcal{P}$: $t_a = t_a - |\mathcal{B}|$, $t_p = t_p - |\mathcal{B}|$ and $\mathcal{P} = \mathcal{P} \setminus \mathcal{B}$. Consequently, the updated t_a, t_p and $\mathcal{P}$ continue to satisfy $t_a + t_p < |\mathcal{P}|$. Below, we will therefore use the fact that $t_a + t_p < |\mathcal{Q}|$, where $\mathcal{Q}$ denotes the relevant set of parties (i.e. the set of parties remaining after possibly expelling a set of identified actively corrupt parties).

Lemma 5. $\mathsf{LRec}^{\alpha,\beta}$ *satisfies the following properties–*

i. Correctness. *Each honest P_i participating in $\mathsf{LRec}^{\alpha,\beta}$ with input L_i as generated by $f_{\mathsf{LSh}}^{\alpha,\beta}(v)$, outputs either v or $\perp$ except with negligible probability.*

ii. Fault-Identification. *If an adversary disrupts the reconstruction of s_j, then $|\mathcal{B}| \geq |\mathcal{Q}| - j$.*

iii. Fairness. *If an adversary disrupts the reconstruction of s_j, then it does not learn s_{j-1}.*

iv. Round Complexity. *It terminates within $\alpha - \beta + 1$ rounds.*

Proof.

Protocol $\mathsf{LRec}^{\alpha,\beta}$

Inputs: Each P_i ($P_i \in \mathcal{Q}$) has input $L_i = \big(\{s_{ji}, o_{ji}\}_{j\in[\alpha,\beta]}, \{c_{jk}\}_{j\in[\alpha,\beta],P_k\in\mathcal{Q}}\big)$.
Output: Secret v or $\perp$ with set $\mathcal{B}$ constituting indices of the identified actively corrupt parties.

- For $j = \alpha$ down to β, P_i does the following round-by-round:
  - Broadcasts (s_{ji}, o_{ji}) and receive (s_{jk}, o_{jk}) from all $P_k \in \mathcal{Q}$ where $k \neq i$.
  - Initialize $\mathsf{Z}_j = i$ and populate Z_j in order to compute s_j as follows:
    - For each $k \neq i$, if commitment c_{jk} opens to s_{jk} via opening o_{jk}, then add k to Z_j.
    - If $|\mathsf{Z}_j| \geq j+1$, interpolate a j-degree polynomial $g_j(x)$ satisfying $g_j(k) = s_{jk}$ for $k \in \mathsf{Z}_j$ and compute $s_j = g_j(0)$. Else output $\perp$, set $\mathcal{B} = \mathcal{Q} \setminus \mathsf{Z}_j$ and terminate.
- Output $v = s_\alpha + \ldots s_\beta$.

Fig. 2: Protocol $\mathsf{LRec}^{\alpha,\beta}$

i. Consider an honest P_i participating with input $L_i = \big(\{s_{ji}, o_{ji}\}_{j\in[\alpha,\beta]}, \{c_{jk}\}_{j\in[\alpha,\beta],P_k\in\mathcal{Q}}\big)$. We observe P_i outputs $v' \neq \{v, \perp\}$ only if at least one of the summands, say $s_j (j \in [\alpha, \beta])$ is incorrectly set. This can happen only if P_i adds at least one index k to Z_j such that P_k sends an incorrect share $s'_{jk} \neq s_{jk}$. This occurs when (s'_{jk}, o'_{jk}) received from P_k is such that c_{jk} opens to s'_{jk} via o'_{jk} but $s'_{jk} \neq s_{jk}$. It now follows directly from the binding of eNICOM that this violation occurs with negligible probability. This completes the proof.

ii. Firstly, it follows from the property of Shamir-secret sharing and binding property of eNICOM that reconstruction of s_j would fail only if $|\mathsf{Z}_j| \leq j$. Next, note that as per the steps in Fig 2, each honest P_i would output $\mathcal{B} = \mathcal{Q} \setminus \mathsf{Z}_j$ if reconstruction of s_j fails. We can thus conclude that $|\mathcal{B}| = |\mathcal{Q}| - |\mathsf{Z}_j| \geq |\mathcal{Q}| - j$.

iii. To prove fairness, we first prove that if an adversary can disrupt the reconstruction of s_j, then it cannot learn s_{j-1} using its eavesdropping power. Since as per the protocol, the honest parties do not participate in the reconstruction of s_{j-1} when they fail to reconstruct s_j, the security of s_{j-1} follows from the information-theoretic security of Shamir-sharing and the statistical security (hiding) of eNICOM.

An adversary can disrupt reconstruction of s_j only if $|\mathsf{Z}_j| \leq j$. It is easy to check that Z_j would constitute the non-actively corrupt parties (honest and purely passive parties) i.e. $\mathcal{Q} \setminus \mathcal{D} \subseteq \mathsf{Z}_j$. Thus, $|\mathcal{Q} \setminus \mathcal{D}| = |\mathcal{Q}| - t_a \leq |\mathsf{Z}_j| \leq j$. Lastly, to maintain $t_a + t_p < |\mathcal{Q}|$, it must hold that $t_p \leq |\mathcal{Q}| - t_a - 1 \leq j - 1$. Thus, the adversary corrupting $t_p \leq j - 1$ parties cannot learn s_{j-1} using its eavesdropping power.

iv. The proof of the round complexity is straightforward. $\mathsf{LRec}^{\alpha,\beta}$ involves reconstruction of summands s_α down to s_β, each of which consumes one round; totalling upto $\alpha - \beta + 1$.

□

4.2 Upper bound for Fair MPC

The key insight for this protocol comes from [HLM13] that builds on an MPC protocol achieving security with abort to compute the function output in $(n-1,1)$-levelled-sharing form, followed by levelled-reconstruction to tackle dynamic corruption. Fairness is brought to the system by relying on the fairness of the levelled-reconstruction. In particular, the adversary is disabled to reconstruct $(i-1)$th summand, as a punitive action, when it disrupts reconstruction of the ith summand for the honest parties. In the marginal case, if the adversary disrupts the MPC protocol for computing the levelled-sharing and does not let the honest parties get their output, we disable it to reconstruct the $(n-1)$th summand itself.

In a (α, β)-levelled-reconstruction, the parameters α and β dictate the round complexity. The closer they are the better round complexity we obtain. The α and β in [HLM13] are $n-2$ apart, shooting the round complexity of reconstruction to $n-1$. We depart from the construction of [HLM13] in two ways to build a $(\lceil\frac{n}{2}\rceil+1)$-round fair protocol. Firstly and prominently, we bring α and β much closer, cutting down $\lfloor\frac{n}{2}\rfloor$ summands from the levelled-secret sharing and bringing down the number of levels to just $n-1-\lfloor\frac{n}{2}\rfloor$ from $n-1$ of [HLM13]. Second, we plug in the round-optimal (2-round) MPC protocol of [GS18, BL18] achieving unanimous abort against malicious majority in the CRS model for computing the levelled-sharing of the output, making overall a $(\lceil\frac{n}{2}\rceil+1)$-round fair protocol. We discuss the first departure in detail below.

Our innovation lies in fixing the best values of α and β without flouting fairness. The value of α and β, in essence determines the indispensable summands that we cannot do without. Every possible *non-zero* threshold for active corruption maps to a crucial summand that the adversary using its corresponding admissible passive threshold cannot learn by itself, whilst the pool of non-disruptive set of parties, i.e. the set of honest and passive parties, can. This unique summand, being the 'soft spot' for the adversary, forces him to cooperate until the reconstruction of the immediate previous summand. As soon as the adversary does so, the honest parties turn self-reliant to compute the output, upholding fairness. We care only about the non-zero possibilities for the threshold of active corruption, as an all-passive adversary holds no power at its disposal to disrupt, leading to robust output reconstruction by all. For the minimum non-zero value of 1 active corruption, the unique summand is s_{n-2} that the adversary cannot learn using its admissible eavesdropping capacity of $n-2$, yet the set of non-disruptive parties, which is of size $n-1$, can. On the other extreme, for the maximum value of $\lceil\frac{n}{2}\rceil-1$, the unique summand is $s_{\lfloor\frac{n}{2}\rfloor}$ that the adversary cannot learn using its admissible eavesdropping capacity of $\lfloor\frac{n}{2}\rfloor$, yet the set of non-disruptive parties, which is of size $\lfloor\frac{n}{2}\rfloor+1$, can. This sets the values of α and β as $n-2$ and $\lfloor\frac{n}{2}\rfloor$ respectively, making the number of crucial summands only $\lceil\frac{n}{2}\rceil-1$. The distance between these two parameters also captures the number of possible corruption scenarios with non-zero active corruption.

In the table below, we display for each admissible adversarial corruption (this set subsumes the crucial summands that we retain), whether the adversary and the set of non-disruptive parties respectively by themselves, can learn the summand, using its maximum eavesdropping capability and putting together their shares respectively. The pattern clearly displays the following feature: irrespective of the corruption scenario that the adversary follows, its maximum power to disrupt and eavesdrop remains one summand apart i.e. if it can disrupt ith summand with its maximum disruptive capability (and fall short of its power for failing the $(i-1)$th one), then its maximum eavesdropping capability does not allow it to learn $(i-1)$th summand by itself.

Table 2: Levelled-reconstruction where $(a = \mathtt{Y}/\mathtt{N}, b = \mathtt{Y}/\mathtt{N})$ under s_i indicates if $\mathcal{A}$ and non-active parties respectively can reconstruct s_i or not ($\mathtt{Y}$ = Yes, $\mathtt{N}$ = No)

$(t_a = \lvert\mathcal{D}\rvert, t_p = \lvert\mathcal{E}\rvert)$	$\lvert\mathcal{P} \setminus \mathcal{D}\rvert$	s_{n-2}	s_{n-3}	s_{n-4}		s_{n-i-1}		$s_{\lfloor n/2 \rfloor+1}$	$s_{\lfloor n/2 \rfloor}$
$(0, n-1)$	n	(Y, Y)	(Y, Y)	(Y, Y)	...	...	...	(Y, Y)	(Y, Y)
$(1, n-2)$	$n-1$	(N, Y)	(Y, Y)	(Y, Y)	...	...	...	(Y, Y)	(Y, Y)
$(2, n-3)$	$n-2$	(N, N)	(N, Y)	(Y, Y)	...	...	...	(Y, Y)	(Y, Y)
...	...	...	...	...	...	...	...	...	...
$(i, n-i-1)$	$n-i$	(N, N)	(N, N)	(N, N)	...	(N, Y)	...	(Y, Y)	(Y, Y)
...	...	...	...	...	...	...	...	...	...
$(\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$	$\lfloor n/2 \rfloor + 1$	(N, N)	(N, N)	(N, N)	...	...	...	(N, N)	(N, Y)

Our fair protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model appears in Fig. 4, where $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ (Fig. 3) denotes the ideal functionality computing $(n-2, \lfloor \frac{n}{2} \rfloor)$-levelled sharing of the output securely with abort. Assumption wise, $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ relies on 2-round maliciously-secure OT in the common random/reference string model (when $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ is realized using protocols of [GS18, BL18]) and eNICOM (used in $\mathsf{LRec}^{\alpha,\beta}$ and instantiated using Pedersen commitment scheme).

Functionality $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$

Input: Receive message (Input, x_i) from P_i $(i \in [n])$. If such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i does not send an input, consider $x_i = \mathtt{abort}$. We require that if $x_i = \mathtt{abort}$, then the adversary corrupts P_i actively.

Output to adversary: If there exists $i \in [n]$ such that $x_i = \mathtt{abort}$, send $(\mathtt{Output}, \bot)$ to all the parties. Else, compute $y = f(x_1, \ldots, x_n)$ and compute $(L_1, \ldots L_n) = f_{\mathsf{LSh}}^{n-2, \lfloor \frac{n}{2} \rfloor}(y)$ (Fig. 1). Send $(\mathtt{Output}, L_i)$ to P_i for every $P_i \in \mathcal{E}$.

Output to honest parties: Receive either $\mathtt{continue}$ or $\mathtt{abort}$ from $\mathcal{A}$. In case of $\mathtt{continue}$, send $(\mathtt{Output}, L_i)$ to each honest P_i, whereas in case of $\mathtt{abort}$ send $(\mathtt{Output}, \bot)$ to all honest parties. We require that an adversary that corrupts no party actively sends $\mathtt{continue}$.

Fig. 3: Ideal Functionality $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$

We state the formal theorem below.

Protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$

Inputs: Party P_j has x_j for $j \in [n]$
Model: $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model (Fig. 3)
Building blocks: Protocol $\mathsf{LRec}^{\alpha,\beta}$ for reconstructing a (α, β)-levelled-shared value (Fig. 2)
Output: $y = f(x_1, \ldots, x_n)$ or $\perp$

Round $1 - 2$**:** Each P_j interacts with $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ with input x_j to compute the function $f_{\mathsf{LSh}}^{n-2,\lfloor \frac{n}{2} \rfloor} \diamond f$ and obtain L_j as the output. If $L_j = \perp$, it outputs $\perp$ and halts.
Round $3 - (\lceil n/2 \rceil + 1)$**:** Each P_j participates in $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$ with input L_j and outputs the outcome of $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$.

Fig. 4: Fair MPC against dynamic-admissible adversary

Theorem 2. *Protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model with n parties satisfies –*

- *Correctness: computes the correct output.*
- *Security: realizes $\mathcal{F}_{\mathsf{fair}}$ (Fig. 15) against a dynamic-admissible $\mathcal{A}$ with threshold (t_a, t_p).*
- *Round complexity: runs in $\lceil n/2 \rceil + 1$ rounds.*

Proof. Correctness of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ follows directly from correctness of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ and $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$ (Lemma 5). Regarding the round complexity, we note that $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ can be realized using the 2-round protocols of [GS18, BL18]. Next, $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$ comprises of $\left(n - 2 - \lfloor \frac{n}{2} \rfloor + 1\right) = \lceil n/2 \rceil - 1$ rounds (Lemma 5). Therefore, the round complexity of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ totals upto $2 + \lceil n/2 \rceil - 1 = \lceil n/2 \rceil + 1$ rounds. □

The security proof of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model appears in Appendix B.1. Standard composition theorems [Can00, Gol04] implies that $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ is secure when access to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ is emulated using 2-round protocols of [GS18, BL18].

4.3 Upper Bound for GOD MPC

At a broad level, robustness is achieved by rerunning our fair protocol as soon as failure occurs which can surface either in the underlying MPC or during reconstruction of any of the summands of the output. Taking inspiration from the player-elimination framework [HMP00, HM01], we maintain a history of deviating/disruptive behaviour across the runs and bar the identified parties from further participating. Such a paradigm calls for sequential runs and brings great challenge when round complexity is the concern. We hit the optimal round complexity banking on several ideas and interesting observations. First, we turn the underlying MPC protocol for computing (α, β)-levelled-sharing of the output to achieve *identifiability* so that any disruptive behaviour can be brought to notice. The recent work of [CGZ20] showed that the 2-round broadcast-only construction of [GS18] can be equipped with identifiability, without inflating the round complexity. Second, we leverage the *function-delayed* property of a modified variant of the protocol of [GS18] (proposed by [ACGJ18]) where the first round messages are made independent of the function to be computed and the number of parties. This enables us to run many parallel instances (specifically $\lceil n/2 \rceil$) of the round 1 in the beginning and run the second round sequentially as and when there is a failure, to compute a new function each time as follows– (a) it hard codes default input for the parties detected to be disruptive so far and (b) the output now is levelled-shared with new thresholds α and β each of which are smaller than the previous run by a function of the number of fresh catch, say δ. The latter brings the most crucial impact on the round complexity. Recall that the distance between α and β that impacts the round complexity, is directly coupled with the number of possible corruption scenarios with non-zero active corruption. Starting with the initial value of $\lceil \frac{n}{2} \rceil - 1$, each catch by δ reduces the number of possible corruption scenarios (with non-zero active corruption) and the distance between α and β by δ.

In the protocol, we maintain a number of dynamic variables which are updated during the run– (a) $\mathcal{L}$: the set of parties not identified to be actively corrupt and thus referred as alive; this set is initialized to $\mathcal{P}$; (b) $\mathcal{C}$: the set of parties identified as actively corrupt; this set initialized to $\emptyset$; (c) $\mathfrak{n}$: the parameter that dictates the number of corruption scenarios as $\lceil \frac{\mathfrak{n}}{2} \rceil$ and the possible corruption cases as $\{(0, \mathfrak{n}-1), \ldots, (\lceil \mathfrak{n}/2 \rceil - 1, \lfloor \mathfrak{n}/2 \rfloor)\}$; this is initialized to n that dictates the initial number of corruption cases as $\lceil \frac{n}{2} \rceil$ and the possible corruption cases as $\{(0, n-1), \ldots, (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)\}$. After every failure and a fresh catch of a set $\mathcal{B}$ of active corruptions, the sets $\mathcal{L}$, $\mathcal{C}$ and $\mathfrak{n}$ are updated as $\mathcal{L} = \mathcal{L} \setminus \mathcal{B}$, $\mathcal{C} = \mathcal{C} \cup \mathcal{B}$ and $\mathfrak{n} = \mathfrak{n} - 2|\mathcal{B}|$. The reduction of $\mathfrak{n}$ by $2|\mathcal{B}|$ denotes counting the reduction for active as well as passive corruptions. For every value of $\mathfrak{n}$, the formula for the total number of corruption scenarios, the values for (α, β) (that speaks about the indispensable summands as discussed in the fair protocol) and the number of corruption scenarios with non-zero active corruption (which denotes the distance between (α, β)) remain the same– namely $\lceil \frac{\mathfrak{n}}{2} \rceil$, $(\mathfrak{n}-2, \lfloor \mathfrak{n}/2 \rfloor)$ and $\lceil \frac{\mathfrak{n}}{2} \rceil - 1$. In the marginal case, $\mathfrak{n}$ becomes either 1 or 2, the former when n is odd and all active corruptions are exposed making $(t_a, t_p) = (0, 0)$ and the latter when n is even and $(t_a, t_p) = (0, 1)$. With no active corruption in $\mathcal{L}$, the Round 2 of the MPC can be run to compute the output itself (instead of its levelled-sharing) robustly in both the marginal cases.

As the protocol follows an inductive behaviour based on $\mathfrak{n}$, to enable better understanding, we present below a snapshot of how the corruption scenarios shrinks after every catch of δ active corruption. The first column indicates a set of possible corruption scenarios, with (t_a, t_p) varying from $(0, \mathfrak{n}-1)$ to $(\lceil \mathfrak{n}/2 \rceil - 1, \lfloor \mathfrak{n}/2 \rfloor)$. If δ cheaters are identified, the first δ rows can simply be discarded as it is established that $t_a \geq \delta$. The number of feasible corruptions is thus slashed by δ. Next, these δ identified cheaters are eliminated, which reduces each (t_a, t_p) of the rows that sustained ($t_a = \delta$ onwards) by δ as shown by column 2. Finally, the column 3 displays column 2 with $\mathfrak{n}$ updated as $\mathfrak{n} - 2\delta$.

The formal description of the protocol $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ appears in Fig 5. While $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ (Fig. 4) was analyzed in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model, such an analysis was not possible for

$\pi_{\mathsf{god}}^{\mathsf{dyn}}$. This is because $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ uses several instances of the 2-round subprotocol π_{idua} such that their Round 1 is always executed (in parallel with Round 1 of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$) but Round 2 is run only when needed; based on the adversarial behaviour during the protocol. Therefore, our round-optimizing tricks do not allow us to analyze this protocol in a hybrid model as it is not possible to substitute these instances of π_{idua} with an ideal functionality (or oracle call).

Assumption wise, $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ relies on 2-round maliciously-secure OT in the common random/reference string model (when π_{idua} is instantiated with function-delayed variant of the protocol of [GS18] satisfying identifiability) and eNICOM (used in $\mathsf{LRec}^{\alpha,\beta}()$ and instantiated using Pedersen commitment scheme).

(t_a, t_p)	(t_a, t_p)	(t_a, t_p)
	after δ cheater identification	after updating $\mathfrak{n} = \mathfrak{n} - 2\delta$
$(0, \mathfrak{n} - 1)$	–	–
$(1, \mathfrak{n} - 2)$	–	–
$\dots$	$\dots$	$\dots$
$(\delta, \mathfrak{n} - \delta - 1)$	$(0, \mathfrak{n} - 2\delta - 1)$	$(0, \mathfrak{n} - 1)$
$(\delta + 1, \mathfrak{n} - \delta - 2)$	$(1, \mathfrak{n} - 2\delta - 2)$	$(1, \mathfrak{n} - 2)$
$\dots$	$\dots$	$\dots$
$(\lceil \mathfrak{n}/2 \rceil - 1, \lfloor \mathfrak{n}/2 \rfloor)$	$(\lceil \mathfrak{n}/2 \rceil - 1 - \delta, \lfloor \mathfrak{n}/2 \rfloor - \delta)$	$(\lceil \mathfrak{n}/2 \rceil - 1, \lfloor \mathfrak{n}/2 \rfloor)$

We now analyze the round-complexity and security of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ below.

Lemma 6. *$\pi_{\mathsf{god}}^{\mathsf{dyn}}$ terminates in $\lceil n/2 \rceil + 1$ rounds.*

Proof. Consider an execution of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ (initialized with $\mathfrak{n} = n$). The outline of the proof is as follows: We give an inductive argument to prove the following - 'If Step 2 is executed with parameter $\mathfrak{n}$, then Step 2 terminates within $\lceil \frac{\mathfrak{n}}{2} \rceil$ rounds'. Assuming this claim holds, it follows directly that during the execution with $\mathfrak{n} = n$, Step 2 would terminate within $\lceil \frac{n}{2} \rceil$ rounds; thereby implying that the round complexity of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ is at most $\lceil \frac{n}{2} \rceil + 1$ (adding the round for Step 1). We now prove the above claim by strong induction on $\mathfrak{n} \geq 1$.

Base Case ($\mathfrak{n} = 1, 2$): It follows directly from description in Fig. 5 that Step 2 terminates in $\lceil \mathfrak{n}/2 \rceil = 1$ round when $\mathfrak{n} = 1, 2$.

Induction Hypothesis ($\mathfrak{n} \leq \ell$): Assume Step 2 terminates in $\lceil \mathfrak{n}/2 \rceil$ rounds for $\mathfrak{n} \leq \ell$.

Induction step ($\mathfrak{n} = \ell + 1$): Consider an execution of Step 2 with parameter $\mathfrak{n} = \ell + 1$. We analyze the following 3 exhaustive scenarios - (1) Suppose neither π_{idua} nor $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$ fails. (2) Suppose π_{idua} aborts. (3) Suppose π_{idua} does not abort but $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$ fails. We show that in each of them, Step 2 terminates within $\lceil \mathfrak{n}/2 \rceil = \lceil \frac{\ell+1}{2} \rceil$ rounds; thereby completing the induction step.

- Suppose neither π_{idua} nor $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$ fails. Then Step 2 involves the following number of rounds– one round (corresponding to Round 2 of π_{idua}) plus the

Protocol $\pi_{\mathsf{god}}^{\mathsf{dyn}}$

Inputs: Party P_i has x_i for $i \in [n]$

Building blocks: (a) Protocol π_{idua} achieving unanimous abort with identifiability (i.e. realizing functionality $\mathcal{F}_{\mathsf{idua}}$, refer Fig 14) against malicious majority and having function-delayed property; (b) Protocol $\mathsf{LRec}^{\alpha,\beta}$ for reconstructing a (α, β)-levelled-shared value (Fig. 2); (c) Function $f_{\mathsf{LSh}}^{\alpha,\beta}$ (Fig. 1).

Output: $y = f(x_1, \ldots, x_n)$

Step 1: P_i runs $\lceil n/2 \rceil$ parallel instances of Round 1 of π_{idua}, each using input x_i and independent randomness. Note that this round is independent of the function to be computed and number of parties. Initialize $k = 1$.

Step 2: Initialize, $\mathcal{L} = \mathcal{P}$, $\mathcal{C} = \emptyset$, $\mathfrak{n} = n$. Let $f^{\mathcal{C}}$ denote the function that is same as f except that the inputs of parties in $\mathcal{C}$ are hard coded with default inputs. P_i executes the following steps:

2.1 If $\mathfrak{n} = 1, 2$, then run Round 2 of π_{idua} (considering kth instance of Round 1) among parties in $\mathcal{L}$ using input x_i to compute $f^{\mathcal{C}}$ and output the output of π_{idua} and terminate. (This corresponds to the case of no active corruptions.)

2.2 Run Round 2 of π_{idua} (considering kth instance of Round 1) among parties in $\mathcal{L}$ using input x_i to compute $f_{\mathsf{LSh}}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor} \diamond f^{\mathcal{C}}$ and obtain L_i. If $L_i = \perp$ and $\mathcal{B}$ is set of parties identified to be corrupt, update $\mathcal{C} = \mathcal{C} \cup \mathcal{B}$, $\mathcal{L} = \mathcal{L} \setminus \mathcal{B}$, $\mathfrak{n} = \mathfrak{n} - 2|\mathcal{B}|$, $k = k + 1$ and repeat this step using updated value of $\mathfrak{n}$. Otherwise, participate in $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}$ with input L_i. If $(\perp, \mathcal{B})$ is the output, then update $\mathcal{L}, \mathcal{C}, \mathfrak{n}, k$ as above and repeat this step using updated value of $\mathfrak{n}$. Otherwise, output the output of $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}$ and terminate.

Fig. 5: Robust MPC against dynamic-admissible adversary

number of rounds in $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}$ (where round complexity of $\mathsf{LRec}^{\alpha,\beta}$ with $\alpha = \mathfrak{n} - 2$ and $\beta = \lfloor \frac{\mathfrak{n}}{2} \rfloor$ is $\alpha - \beta + 1 = \mathfrak{n} - 2 - \lfloor \frac{\mathfrak{n}}{2} \rfloor + 1 = \lceil \mathfrak{n}/2 \rceil - 1$). This totals upto $1 + (\lceil \mathfrak{n}/2 \rceil - 1) = \lceil \mathfrak{n}/2 \rceil$.

- Suppose π_{idua} aborts. Then $\mathcal{B}$ must comprise of at least one active party, implying that $\delta \geq 1$, where $\delta = |\mathcal{B}|$ and subsequently $\mathfrak{n}$ is updated to $\mathfrak{n} = (\mathfrak{n} - 2\delta) \leq (\ell + 1 - 2) = (\ell - 1)$. Note that Step 2 now involves following number of rounds– 1 (for Round 2 of π_{idua}) + number of rounds in which Step 2 terminates when re-run with updated parameter $\mathfrak{n}$ i.e. $\lceil \mathfrak{n}/2 \rceil$ by induction hypothesis. Thus, the total number of rounds in Step 2 is $(1 + \lceil \mathfrak{n}/2 \rceil) \leq (1 + \lceil \frac{\ell-1}{2} \rceil) = \lceil \frac{\ell+1}{2} \rceil$.
- Suppose π_{idua} does not abort but reconstruction $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}$ fails. Say adversary disrupts reconstruction of summand $s_{\mathfrak{n}-r}$ in Round r of Step 2 (Round $r-1$ of $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \mathfrak{n}/2 \rfloor}$), where $r \in [2, \lceil \mathfrak{n}/2 \rceil]$. It follows from fault identification property of Lemma 5 that $|\mathcal{B}| \geq |\mathcal{L}| - (\mathfrak{n} - r) \geq r$ (since $|\mathcal{L}| \geq \mathfrak{n}$ always holds). Consequently, $\delta = |\mathcal{B}| \geq r$ and updated parameter $\mathfrak{n} = \mathfrak{n} - 2\delta \leq \ell + 1 - 2r$. We now analyze the round complexity. Note that Step 2 involves following number of rounds– r (Reconstruction failed in Round $r \geq 2$ of Step 2 run with $\mathfrak{n} = \ell + 1$) + number of rounds in which Step 2 terminates when re-run

with updated parameter $\mathfrak{n}$ i.e. $\lceil \mathfrak{n}/2 \rceil$ by induction hypothesis. Thus total number of rounds in Step 2 is $(r + \lceil \mathfrak{n}/2 \rceil) \leq (r + \lceil \frac{\ell+1-2r}{2} \rceil) = \lceil \frac{\ell+1}{2} \rceil$.

We point that induction hypothesis for $\mathfrak{n} = \mathfrak{n} - 2\delta$ with $\delta \geq 1$ can be applied as $\mathfrak{n} \geq 1$ holds always in $\pi^{\mathsf{dyn}}_{\mathsf{god}}$ due to the following: the maximal value of δ is $\lceil \mathfrak{n}/2 \rceil - 1$ i.e. the maximum possible number of actively corrupt parties. This completes the proof. □

Theorem 3. *Assuming the presence of a 2-round protocol π_{idua} realizing functionality $\mathcal{F}_{\mathsf{idua}}$ (Fig 14) against malicious majority and having function-delayed property; protocol $\pi^{\mathsf{dyn}}_{\mathsf{god}}$ with n parties satisfies–*

- *Correctness: computes the correct output.*
- *Security: realizes $\mathcal{F}_{\mathsf{god}}$ (Fig. 16) against a dynamic-admissible $\mathcal{A}$ with threshold (t_a, t_p).*
- *Round complexity: runs in $\lceil n/2 \rceil + 1$ rounds.*

Proof. Correctness of $\pi^{\mathsf{dyn}}_{\mathsf{god}}$ follows directly from correctness of π_{idua} and correctness of $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}$ (Lemma 5). Round complexity follows from Lemma 6. □
The formal security proof appears in Appendix B.2.

5 Lower Bounds for Boundary Corruption

In this section, we present two lower bounds for MPC protocol tolerating boundary-admissible adversaries and in the presence of CRS and PKI setup. Notably, both lower bounds extend to a model with arbitrary correlated randomness.

Recall that the boundary adversary is restricted to corruption scenarios either $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$. We show that *three* and *four* rounds are necessary to achieve fairness and GOD respectively against a boundary-admissible adversary. It is to be noted that GOD is the de facto notion achieved in the pure passive corruption setting of $(t_a, t_p) = (0, n-1)$.

5.1 Impossibility of 3-round Robust MPC

In this section, we show that it is impossible to design a 3-round robust MPC protocol against boundary-admissible adversary with threshold (t_a, t_p) assuming both CRS and PKI. Notably, this lower bound is indeed surprising as the individual security guarantees translate to GOD against malicious-minority [GLS15] and passive-majority [GS18, BL18] for odd n (as $t_a = t_p$ wrt $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$), both of which are known to be attainable in just 2 rounds in the presence of CRS and PKI. Furthermore, it turns out interestingly that this lower bound does not hold against a boundary-admissible adversary with $t_a \leq 1$ (i.e. boundary adversary corrupting with either $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$), and can be circumvented for this special case. In fact, we demonstrate a 3-round robust protocol in Section 6.3, against this special-case boundary-admissible adversary.

We first present a high-level description of the proof. Towards a contradiction, we assume that there exists a 3-round 5-party protocol π computing a common output function f that achieves GOD against a boundary-admissible adversary, who may corrupt either with parameters $(t_a, t_p) = (2, 2)$ or $(t_a, t_p) = (0, 4)$. Since our lower bound argument demands the presence of at least two active corruptions, we choose the minimal n for which this holds i.e. $n = 5$. The argument can also be extended for $n > 5$ as elaborated later.

The argument involves three adversarial strategies $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, where $\mathcal{A}_i$ is launched in an execution $\varSigma_i$ of protocol π. While $\mathcal{A}_1, \mathcal{A}_2$ involve the case of active corruption of $\{P_1\}$ and $\{P_1, P_2\}$ respectively, $\mathcal{A}_3$ deals with the strategy of pure passive corruption of $\{P_1, P_3, P_4, P_5\}$. The executions are assumed to be run for the same input tuple $(x_1, x_2, x_3, x_4, x_5)$ and the same random inputs $(r_1, r_2, r_3, r_4, r_5)$ of the parties. Let $\widetilde{x_i}$ denote the default input of P_i. (Same random inputs are considered for simplicity and without loss of generality. The same arguments hold for distribution ensembles as well.) First, when $\mathcal{A}_1$ is launched in $\varSigma_1$ we conclude that the output $\widetilde{y}$ at the end of the execution should be based on default input of P_1 and actual inputs of the remaining parties i.e. $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$. Next, strategy $\varSigma_2$ involving actively corrupt $\{P_1, P_2\}$ is designed such that corrupt P_2 obtains the same view in $\varSigma_2$ as an honest P_2 in $\varSigma_1$ and therefore computes the output $\widetilde{y}$ at the end of $\varSigma_2$. Lastly, a carefully designed strategy $\mathcal{A}_3$ by semi-honest parties $\{P_1, P_3, P_4, P_5\}$ allows $\mathcal{A}$ to obtain $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$, in addition to the correct output i.e. $y = f(x_1, x_2, x_3, x_4, x_5)$ at the end of execution $\varSigma_3$. This is a contradiction as it violates the security of π and can explicitly breach the privacy of honest P_2. This completes the proof overview which we formalize below.

Theorem 4. *Assume parties have access to pairwise-private and broadcast channels, and a setup that includes* CRS *and* PKI*. Then, there exist functions f for which there is no 3-round protocol computing f that achieves guaranteed output delivery against boundary-admissible adversary.*

Proof. We prove the theorem for $n = 5$ parties. Let $\mathcal{P} = \{P_1, \dots P_5\}$ denote the set of parties, where the adversary $\mathcal{A}$ may corrupt either with parameters $(t_a, t_p) = (2, 2)$ or $(t_a, t_p) = (0, 4)$. Here, the corruption scenarios translate to upto 2 active corruptions or upto 4 pure passive corruptions. We prove the theorem by contradiction. Suppose there exists a 3-round protocol π computing a common output function f that achieves GOD against such a boundary-admissible adversary. We assume that the communication done in Round 2 and Round 3 of π is via broadcast alone. This holds without loss of generality since the parties can engage in point-to-point communication by exchanging random pads in the first round and then use these random pads to unmask later broadcasts.

We use the following notation: Let $\mathsf{p}^1_{i \to j}$ denote the pairwise communication from P_i to P_j in round 1 and b^r_i denotes the broadcast by P_i in round r, where

$r \in [3], \{i, j\} \in [5]$. These values may be function of CRS and the PKI setup as per the protocol specifications. Let V_i^ℓ denotes the view of party P_i at the end of execution Σ_ℓ ($\ell \in [3]$) of π. Here, view of P_i includes its input x_i, randomness r_i, the messages received during π and the knowledge related to CRS and PKI setup. Below we describe the strategies $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, where $\mathcal{A}_i$ is launched in an execution Σ_i of protocol π. The executions are assumed to be run for the same input tuple $(x_1, x_2, x_3, x_4, x_5)$ and the same random inputs $(r_1, r_2, r_3, r_4, r_5)$ of the parties.

$\mathcal{A}_1$: $\mathcal{A}$ corrupts $\{P_1\}$ actively here. P_1 behaves honestly in Round 1 and simply remains silent in Round 2 and Round 3.

$\mathcal{A}_2$: $\mathcal{A}$ corrupts $\{P_1, P_2\}$ actively here. The active misbehavior of P_1 is same as in $\mathcal{A}_1$ i.e. P_1 behaves honestly in Round 1 and stops communicating thereafter. On the other hand, P_2 participates honestly upto Round 2 and remains silent in Round 3.

$\mathcal{A}_3$: $\mathcal{A}$ corrupts $\{P_1, P_3, P_4, P_5\}$ passively here. The semi-honest parties behave as per protocol specification throughout the execution Σ_3 to obtain the correct output. The passive strategy of $\{P_1, P_3, P_4, P_5\}$ is to ignore the Round 3 message from honest P_2 and locally compute the output based on the scenario of execution Σ_2 i.e. imagining that P_1 stopped after Round 1 and P_2 stopped after Round 2.

We present a table depicting the views of the parties in executions Σ_1 and Σ_2 in Table 3. Here $\overline{\mathsf{b}_i^3}$ for $i \in \{2, 3, 4, 5\}$ denotes the message broadcast by honest P_i (as per its next-message function) in Round 3 in case P_1 behaves honestly in Round 1 but is silent in Round 2. The views of parties in Σ_3 which is as per honest execution (since it involves only purely passive corruptions) appears in Table 4.

Table 3: Views of P_1, P_2, P_3, P_4, P_5 in Σ_1 and Σ_2

	Σ_1					Σ_2				
	V_1^1	V_2^1	V_3^1	V_4^1	V_5^1	V_1^2	V_2^2	V_3^2	V_4^2	V_5^2
Input	(x_1, r_1)	(x_2, r_2)	(x_3, r_3)	(x_4, r_4)	(x_5, r_5)	(x_1, r_1)	(x_2, r_2)	(x_3, r_3)	(x_4, r_4)	(x_5, r_5)
R1	$\mathsf{p}^1_{2\to1}$, $\mathsf{p}^1_{3\to1}$, $\mathsf{p}^1_{4\to1}$, $\mathsf{p}^1_{5\to1}$, b^1_2, b^1_3, b^1_4, b^1_5	$\mathsf{p}^1_{1\to2}$, $\mathsf{p}^1_{3\to2}$, $\mathsf{p}^1_{4\to2}$, $\mathsf{p}^1_{5\to2}$, b^1_1, b^1_3, b^1_4, b^1_5	$\mathsf{p}^1_{1\to3}$, $\mathsf{p}^1_{2\to3}$, $\mathsf{p}^1_{4\to3}$, $\mathsf{p}^1_{5\to3}$, b^1_1, b^1_2, b^1_4, b^1_5	$\mathsf{p}^1_{1\to4}$, $\mathsf{p}^1_{2\to4}$, $\mathsf{p}^1_{3\to4}$, , $\mathsf{p}^1_{5\to4}$ b^1_1, b^1_2, b^1_3, b^1_5	$\mathsf{p}^1_{1\to5}$, $\mathsf{p}^1_{2\to5}$, $\mathsf{p}^1_{3\to5}$, , $\mathsf{p}^1_{4\to5}$ b^1_1, b^1_2, b^1_3, b^1_4	$\mathsf{p}^1_{2\to1}$, $\mathsf{p}^1_{3\to1}$, $\mathsf{p}^1_{4\to1}$, $\mathsf{p}^1_{5\to1}$, b^1_2, b^1_3, b^1_4, b^1_5	$\mathsf{p}^1_{1\to2}$, $\mathsf{p}^1_{3\to2}$, $\mathsf{p}^1_{4\to2}$, $\mathsf{p}^1_{5\to2}$, b^1_1, b^1_3, b^1_4, b^1_5	$\mathsf{p}^1_{1\to3}$, $\mathsf{p}^1_{2\to3}$, $\mathsf{p}^1_{4\to3}$, $\mathsf{p}^1_{5\to3}$, b^1_1, b^1_2, b^1_4, b^1_5	$\mathsf{p}^1_{1\to4}$, $\mathsf{p}^1_{2\to4}$, $\mathsf{p}^1_{3\to4}$, $\mathsf{p}^1_{5\to4}$, b^1_1, b^1_2, b^1_3, b^1_5	$\mathsf{p}^1_{1\to5}$, $\mathsf{p}^1_{2\to5}$, $\mathsf{p}^1_{3\to5}$, $\mathsf{p}^1_{4\to5}$, b^1_1, b^1_2, b^1_3, b^1_4
R2	b^2_2, b^2_3, b^2_4, b^2_5	$-$, b^2_3, b^2_4, b^2_5	$-$, b^2_2, b^2_4, b^2_5	$-$, b^2_2, b^2_3, b^2_5	$-$, b^2_2, b^2_3, b^2_4	b^2_2, b^2_3, b^2_4, b^2_5	$-$, b^2_3, b^2_4, b^2_5	$-$, b^2_2, b^2_4, b^2_5	$-$, b^2_2, b^2_3, b^2_5	$-$, b^2_2, b^2_3, b^2_4
R3	$\overline{\mathsf{b}^3_2}$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $\overline{\mathsf{b}^3_2}$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $\overline{\mathsf{b}^3_2}$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_5}$	$-$, $\overline{\mathsf{b}^3_2}$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$	$-$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $-$, $\overline{\mathsf{b}^3_4}$, $\overline{\mathsf{b}^3_5}$	$-$, $-$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_5}$	$-$, $-$, $\overline{\mathsf{b}^3_3}$, $\overline{\mathsf{b}^3_4}$

Table 4: Views of P_1, P_2, P_3, P_4, P_5 in Σ_3

	Σ_3				
	V_1^1	V_2^1	V_3^1	V_4^1	V_5^1
Input	(x_1, r_1)	(x_2, r_2)	(x_3, r_3)	(x_4, r_4)	(x_5, r_5)
R1	$\mathsf{p}_{2\to1}^1$, $\mathsf{p}_{3\to1}^1$ $\mathsf{p}_{4\to1}^1$, $\mathsf{p}_{5\to1}^1$ b_2^1, b_3^1, b_4^1, b_5^1	$\mathsf{p}_{1\to2}^1$, $\mathsf{p}_{3\to2}^1$, $\mathsf{p}_{4\to2}^1$, $\mathsf{p}_{5\to2}^1$ b_1^1, b_3^1, b_4^1, b_5^1	$\mathsf{p}_{1\to3}^1$, $\mathsf{p}_{2\to3}^1$, $\mathsf{p}_{4\to3}^1$, $\mathsf{p}_{5\to3}^1$ b_1^1, b_2^1, b_4^1, b_5^1	$\mathsf{p}_{1\to4}^1$, $\mathsf{p}_{2\to4}^1$, $\mathsf{p}_{3\to4}^1$, $\mathsf{p}_{5\to4}^1$ b_1^1, b_2^1, b_3^1, b_5^1	$\mathsf{p}_{1\to5}^1$, $\mathsf{p}_{2\to5}^1$, $\mathsf{p}_{3\to5}^1$, $\mathsf{p}_{4\to5}^1$ b_1^1, b_2^1, b_3^1, b_4^1
R2	b_2^2, b_3^2, b_4^2, b_5^2	b_1^2, b_3^2, b_4^2, b_5^2	b_1^2, b_2^2, b_4^2, b_5^2	b_1^2, b_2^2, b_3^2, b_5^2	b_1^2, b_2^2, b_3^2, b_4^2
R3	b_2^3, b_3^3, b_4^3, b_5^3	b_1^3, b_3^3, b_4^3, b_5^3	b_1^2, b_2^3, b_4^3, b_5^3	b_1^3, b_2^3, b_3^3, b_5^3	b_1^3, b_2^3, b_3^3, b_4^3

We now present a sequence of lemmas to complete the proof. Let $\mathsf{p} = 1 - \mathtt{negl}(\kappa)$ denote the overwhelming probability with which security of π holds, where the probability is defined over the choice of setup and the random coins used by the parties.

Lemma 7. *At the end of Σ_1, parties compute output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$ with overwhelming probability, where $\widetilde{x_1}$ denotes the default input of P_1.*

Proof. Firstly, since Σ_1 involves active behavior only by P_1, it follows directly from security of π (which holds with overwhelming probability p) that the output computed at the end of Σ_1, say y' should be based on actual inputs x_i for $i \in \{2, 3, 4, 5\}$. Now, there are two possibilities with respect to input of P_1 i.e. y' is based on either x_1 (i.e. the input used by P_1 in Round 1 of Σ_1) or $\widetilde{x_1}$ (default input). In case of the latter, the lemma holds directly. We now assume the former for contradiction.

Suppose the output y' (computed with overwhelming probability p) is based on x_1 rather than $\widetilde{x_1}$. Since P_1 stops communicating after Round 1, we can conclude that the combined views of $\{P_2, P_3, P_4, P_5\}$ must suffice to compute the output $y' = f(x_1, \ldots, x_5)$ at the end of Round 1 itself. If this holds, we argue that π cannot be secure as follows: Suppose π is such that when all parties participate honestly in Round 1, the combined view of $\{P_2, P_3, P_4, P_5\}$ suffices to compute the output at the end of Round 1 itself, with overwhelming probability p. Then, in an execution of π, an adversary corrupting $\{P_2, P_3, P_4, P_5\}$ purely passively (corresponding to $(t_a, t_p) = (0, 4)$) can learn the output with overwhelming probability p, on various inputs of its choice, keeping x_1 fixed. This residual attack breaches privacy of honest P_1 (A concrete example of such an f appears at the end of this section). We have thus arrived at a contradiction. This completes the proof that y' must be based on $\widetilde{x_1}$, rather than x_1 and consequently $y' = \widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$ must be the output computed at the end of Σ_1 with overwhelming probability p. □

Lemma 8. *At the end of Σ_2, parties compute output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$ with overwhelming probability, where $\widetilde{x_1}$ denotes the default input of P_1.*

Proof. Recall that $\mathcal{A}_2$ is similar to $\mathcal{A}_1$ involving active P_1, except that P_2 is active as well with the strategy of behaving honestly upto Round 2 and remaining silent in Round 3. Since the executions Σ_1 and Σ_2 proceed identically upto Round 2, it is easy to check that the view of corrupt P_2 in Σ_2 is same as honest P_2 in Σ_1 (refer to Table 3). It now follows directly from Lemma 7 that P_2 can learn the output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$ with overwhelming probability p. By security of π (which holds with overwhelming probability p) computing the common output function f, it must hold that when P_2 obtains $\widetilde{y}$, all parties must output $\widetilde{y}$ at the end of Σ_2 with overwhelming probability $\mathsf{p} \times \mathsf{p} = \mathsf{p}^2$. □

Lemma 9. *The combined view of parties $\{P_3, P_4, P_5\}$ at the end of Round 2 of Σ_2 suffices to compute the output of Σ_2 i.e. $\widetilde{y}$ with overwhelming probability.*

Proof. We note that as per $\mathcal{A}_2$, both $\{P_1, P_2\}$ do not communicate in Round 3; implying that the combined view of honest parties $\{P_3, P_4, P_5\}$ at the end of Round 2 of Σ_2 must suffice to compute the output of Σ_2 i.e. $\widetilde{y}$ with overwhelming probability p^2 (Lemma 8). □

Lemma 10. *An adversary executing strategy $\mathcal{A}_3$ obtains the value $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$, in addition to the correct output $y = f(x_1, x_2, x_3, x_4, x_5)$ at the end of Σ_3, with overwhelming probability.*

Proof. Firstly, Σ_3 must lead to computation of correct output i.e. $y = f(x_1, x_2, x_3, x_4, x_5)$ by all parties with overwhelming probability p since $\mathcal{A}_3$ involves only semi-honest corruptions. Next, it is easy to check from Tables 3 and 4 that the combined view of adversary corrupting $\{P_1, P_3, P_4, P_5\}$ passively at the end of Round 2 of Σ_3 subsumes the combined view of honest parties $\{P_3, P_4, P_5\}$ at the end of Round 2 of Σ_2. It now follows directly from Lemma 9 that the adversary can obtain the output $\widetilde{y}$ as well with overwhelming probability p^2.

In more detail, $\mathcal{A}$ launching $\mathcal{A}_3$ in Σ_3 can compute the output as per the scenario of Σ_2 as follows- Let $\overline{\mathsf{b}_i^3}$ for $i \in \{2, 3, 4, 5\}$ denotes the message broadcast by honest P_i (as per its next-message function) in Round 3 in case P_1 behaves honestly in Round 1 but is silent in Round 2. Locally compute $\{\overline{\mathsf{b}_3^3}, \overline{\mathsf{b}_4^3}, \overline{\mathsf{b}_5^3}\}$ ($\overline{\mathsf{b}_i^3}$ is a function of P_i's ($i \in \{3, 4, 5\}$) view at the end of Round 2) by imagining that P_1 did not send Round 2 message and compute $\widetilde{y}$ by ignoring the message sent by honest P_2 in Round 3. Thus, by following strategy $\mathcal{A}_3$, $\mathcal{A}$ obtains multiple evaluations of f i.e. both y and $\widetilde{y}$ with overwhelming probability, which violates the security of π. (We give a concrete example of such an f below that breaches privacy of honest P_2.) This completes the proof of the lemma. □

Thus, we have arrived at a contradiction to our assumption that π is secure. While the above proof was shown specifically for $n = 5$, it can be extended to any $n > 5$ in the following natural manner: The strategies $\mathcal{A}_1, \mathcal{A}_2$ remain the same (feasible as at least two active corruptions are allowed when $n > 5$) and let us conclude that the combined view of $\{P_3, P_4 \ldots, P_n\}$ at the end of Round 2 suffices to compute $\widetilde{y} = f(\widetilde{x_1}, x_2 \ldots, x_n)$ with overwhelming probability. Accordingly, strategy $\mathcal{A}_3$ involving passive corruption of $\{P_1, P_3, P_4 \ldots, P_n\}$ would lead to the adversary obtaining multiple evaluations of the function with overwhelming probability leading to the final contradiction. This completes the proof of Theorem 4. □

Next, we give a concrete example of f to elaborate on how the residual attack can be executed to breach privacy.

Concrete Example of f: Let $f(x_1, x_2, x_3, x_4, x_5)$ with $x_1 = (\alpha, \beta), x_2 = (b, m_0, m_1)$ (where α, β, b are single bit values) and $x_3 = x_4 = x_5 = \perp$ be defined as below for P_i's input x_i:

$$f(x_1, x_2, x_3, x_4, x_5) = \begin{cases} m_\alpha & \text{if } b = 0 \\ m_{\alpha \oplus \beta} & \text{otherwise} \end{cases}$$

Using this function f, we describe explicitly how multiple evaluations of f breaches privacy of P_1 and P_2 in the argument of Lemma 7 and Lemma 9 respectively. Consider the adversary corrupting $\{P_2, P_3, P_4, P_5\}$ passively $((t_a, t_p) = (0, 4))$ that can learn the output on various inputs of its choice, keeping x_1 fixed (in Lemma 7). By locally plugging in inputs $b = 0$ and $b = 1$ on behalf of passive P_2, it is easy to check that the adversary can learn both α and β. This violates privacy of honest P_1 as its input β is never revealed as per the ideal functionality. Next, consider the adversary of Lemma 9 corrupting $\{P_1, P_3, P_4, P_5\}$ who obtains both $y = f(x_1, x_2, x_3, x_4, x_5)$ and $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$. We claim this breaches privacy of honest P_2 as follows: As per the ideal functionality, the adversary would learn exactly only one among m_0, m_1. Next, suppose the default value of $\widetilde{x_1} = (0, 0)$. Then by participating in Σ_3 with input $x_1 = (1, 0)$, the adversary would obtain both $y = m_1$ and $\widetilde{y} = m_0$ (irrespective of b) which compromises the security of honest P_2's input.

Before concluding this section, we give quick intuition of why the above lower bound argument does not hold when malicious corruption $t_a \leq 1$. Note that the strategy $\mathcal{A}_3$ carried out by the adversary corrupting $\{P_1, P_3, P_4, P_5\}$ purely passively was feasible only since the output on default input of P_1 could be computed without any dependency on honest P_2's message in Round 3. Had it been the case that honest P_2's Round 3 message was crucial for output computation, then the semi-honest parties $\{P_1, P_3, P_4, P_5\}$ would have obtained only the output on the combination of actual inputs and would be unable to breach security. Tracing back, recall that the partnership of malicious $\{P_1, P_2\}$ together in $\mathcal{A}_2$ was crucial in implying this non-dependency on Round 3 message of P_2 (which led us to the conclusion of view of $\{P_3, P_4, P_5\}$ being sufficient to compute output on P_1's default input). It is thereby evident that without such a partnership of two malicious parties, it would not be possible to arrive at such a contradiction. This intuition is further substantiated by our 3-round upper bound achieving GOD in case of single active corruption (Section 6.3).

5.2 Impossibility of 2-round Fair MPC

We begin with the observation that the existing 3-round lower bounds of [GIKR02, GLS15, PR18a] for fair malicious-minority MPC do not carry over in our setting. The lower bound of both [GIKR02, GLS15] break down when the parties have access to a PKI (as acknowledged/demonstrated in their work). The result of [PR18a], assuming access to pairwise-private and broadcast channels, also breaks down when parties have access to a PKI. The proof, originally given without the mention of CRS, seems to withstand a CRS. The proof approach of

[PR18a] is via contradiction i.e. derives a series of implications assuming that 2-round fair MPC protocol π exists and eventually builds up to a contradiction. A crucial lemma in their proof (Lemma 24 in their full version [PR18b]) states that π must be such that a single party, say P_1, is able to compute the output at the end of Round 1. The argument for this claim relies on the fact that (a) the adversary's communication stops after Round 1 and (b) the Round 2 messages of honest parties do not hold any potential useful information to aid P_1's output computation. Roughly speaking, (b) follows since the honest party's messages are fully determined by the information available to P_1 at the end of Round 1 itself and can therefore be locally computed by P_1. This information includes the broadcast communication by the adversary in Round 1. While the above argument regarding (b) holds in the plain model and even in the presence of public setup such as CRS, it does not hold in the presence of private setup like PKI. In this case, an honest party may hold some private information unknown to P_1 at the end of Round 1, such as the decryption of the adversary's Round 1 broadcast using its exclusive secret key; which may aid in output computation by P_1. Consequently, this claim of [PR18a] and their proof are not resilient to the presence of PKI.

Before presenting our lower bound formally, we present the proof sketch. Towards a contradiction, we assume π is a 3-party protocol computing f that achieves fairness against a boundary-admissible adversary $\mathcal{A}$. We first exploit fairness of π to conclude that the combined view of a set of 2 parties suffices for output computation at the end of Round 1. (Here, view of P_i includes its input x_i, its randomness r_i, the messages received during π and the knowledge related to CRS and PKI setup.) Next, considering a strategy where the adversary $\mathcal{A}$ corrupts this set of 2 parties purely passively leads us to conclude that $\mathcal{A}$ can compute the output at the end of Round 1 itself; leading upto a final contradiction.

Theorem 5. *There exist functions f for which there is no 2-round n-party MPC protocol that achieves fairness against boundary-admissible adversary, in a setting with pairwise-private and broadcast channels, and a setup that includes* CRS *and* PKI.

Proof. We prove the theorem for $n = 3$ parties, where boundary-admissible adversary $\mathcal{A}$ chooses corruption parameters either $(t_a, t_p) = (1, 1)$ or $(t_a, t_p) = (0, 2)$. Here, the corruption scenarios translate to either upto 1 active corruption or upto 2 purely passive corruptions. Let $\{P_1, P_2, P_3\}$ denote the set of parties with P_i having input x_i. Suppose by contradiction, π is an MPC protocol computing f that achieves fairness against $\mathcal{A}$. To be more specific, π is fair if $(t_a, t_p) = (1, 1)$ and achieves GOD otherwise (as GOD is the de-facto security guarantee in case of no active corruptions i.e. $(t_a, t_p) = (0, 2)$).

We now present the sequence of claims. Let $\mathtt{p} = 1 - \mathtt{negl}(\kappa)$ denote the overwhelming probability with which security of π holds, where the probability is defined over the choice of setup and the random coins used by the parties.

Lemma 11. *Protocol π must be such that the combined view of $\{P_2, P_3\}$ at the end of Round 1 suffices for output computation with overwhelming probability.*

Proof. The proof of the lemma is straightforward. Assume $\mathcal{A}$ corrupting P_1 actively (with $(t_a, t_p) = (1, 1)$) with the following strategy: P_1 behaves honestly in Round 1 and simply remains silent in Round 2. It is easy to check that P_1 would obtain the output with overwhelming probability p, due to correctness of π, as he receives the entire protocol communication as per honest execution. Since π is fair (with overwhelming probability p), when P_1 obtains the output, the honest parties $\{P_2, P_3\}$ must also obtain the output at the end of π with overwhelming probability $\mathsf{p} \times \mathsf{p} = \mathsf{p}^2$. Note that $\{P_2, P_3\}$ compute the output without P_1's communication in Round 2. Thus, we conclude that the combined view of $\{P_2, P_3\}$ at the end of Round 1 suffices for output computation with overwhelming probability p^2. □

Lemma 12. *There exists an adversarial strategy such that the adversary obtains the output at the end of Round 1, with overwhelming probability.*

Proof. The proof follows directly from Lemma 11– $\mathcal{A}$ corrupting $\{P_2, P_3\}$ purely passively ($(t_a, t_p) = (0, 2)$) would obtain the output at the end of Round 1, with overwhelming probability p^2. □

Lemma 13. *There exists an adversarial strategy that enables $\mathcal{A}$ to breach privacy of the protocol π with overwhelming probability.*

Proof. It is implied from Lemma 12 that $\mathcal{A}$ corrupting $\{P_2, P_3\}$ purely passively can obtain multiple evaluations of the function f with overwhelming probability, by locally plugging in different values for $\{x_2, x_3\}$ while honest P_1's input x_1 remains fixed. This 'residual function attack' violates privacy of P_1. We refer to the argument in Lemma 4 for a concrete example. □

We have arrived at a contradiction, concluding the proof of Theorem 5. It is easy to check that this argument can be extended for higher values of n. □

6 Upper bounds for Boundary Corruption

In this section, we describe three upper bounds with respect to the boundary-admissible adversary $\mathcal{A}$ with threshold (t_a, t_p). We first present a robust upper bound in 4 rounds for the general case. Next, we present a 3-round robust protocol for the special case of single active corruption, which circumvents our lower bound of Section 5.1. Finally, we present our fair 3-round upper bound that can be arrived at by simplifying the robust general-case construction. Note that even the fair construction is robust in the corruption scenario of no active corruptions i.e. $(t_a, t_p) = (0, n-1)$. The security guarantees differ only in case of corruption scenario involving malicious corruptions. All the above three constructions are round-optimal, following our lower bound results of Section 5.1 and 5.2. We start with a building block commonly used across all our constructs.

6.1 Authenticated Secret Sharing

We introduce the primitive of Authenticated Secret Sharing [IKP+16, IKK+11] used in our upper bounds against the boundary-admissible $\mathcal{A}$.

Definition 4 (α-authenticated sharing). *A value v is said to be α-authenticated-shared amongst a set of parties $\mathcal{P}$ if every honest or passively corrupt party P_i in $\mathcal{P}$ holds S_i as produced by $f^{\alpha}_{\mathsf{ASh}}(v)$ given in Fig.6.*

Function $f^{\alpha}_{\mathsf{ASh}}(v)$

1. *α shamir-sharing of secret v:* Choose random $a_1, a_2 \dots a_\alpha \in \mathbb{F}$, where $\mathbb{F}$ denotes a finite field. Build the α-degree polynomial $A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{\alpha-1} x^{\alpha-1} + a_\alpha x^\alpha$, where $a_0 = v$. Let $\mathsf{sh}_i = A(i)$ for $i \in [n]$.
2. *Authentication of shares:* For all $i, j \in [n]$, choose random one-time message-authentication codes (MAC) [Gol04] keys $k_{ij} \in \{0,1\}^\kappa$ and compute $\mathsf{tag}_{ij} = \mathsf{Mac}_{k_{ij}}(\mathsf{sh}_i)$.
3. Output $S_i = \big(\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big)$ for $i \in [n]$.

Fig. 6: Authenticated secret-sharing

In our upper bounds, the function $f^{\alpha}_{\mathsf{ASh}}$ is realized via MPC protocols. The reconstruction will be done via protocol ARec^α (Fig 7) amongst the parties. We prove the relevant properties below:

Protocol ARec^α

Inputs: P_i participates with $S_i = \big(\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big)$
Output: Secret v'

Each P_i does the following:

1. Broadcast $\big(\mathsf{sh}_i, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big)$ and receive $\big(\mathsf{sh}'_j, \mathsf{tag}'_{ji}\big)$ from $j \neq i$.
2. Each P_i outputs v' as follows:.
   - Initialize Val to $\{i\}$. For $j \neq i$, if $\mathsf{Mac}_{k_{ji}}(\mathsf{sh}'_j) = \mathsf{tag}'_{ji}$, set $\mathsf{sh}_j = \mathsf{sh}'_j$ and add j to Val; else set $\mathsf{sh}_j = \perp$.
   - If $|\mathsf{Val}| \geq \alpha + 1$, interpolate a α degree polynomial $A'(x)$ satisfying $A'(\gamma) = \mathsf{sh}_\gamma$ for $\gamma \in \mathsf{Val}$. Output $\perp$ if the above fails, else output $v' = A'(0)$.

Fig. 7: Protocol for Reconstruction of an authenticated-secret

Lemma 14. *The pair $(f^{\alpha}_{\mathsf{ASh}}, \mathsf{ARec}^\alpha)$ satisfies the following:*

i. Privacy. *For all $v \in \mathbb{F}$, the output $(S_1, \dots, S_n) \leftarrow f^{\alpha}_{\mathsf{ASh}}(v)$ satisfies the following– $\forall \{i_1, \dots i_{\alpha'}\} \subset [n]$ with $\alpha' \leq \alpha$, the distribution of $\{S_{i_1}, \dots, S_{i_{\alpha'}}\}$ is statistically independent of v.*

ii. Correctness. *For all $v \in \mathbb{F}$, the value v' output by all honest parties at the end of $\mathsf{ARec}^{\alpha}(S'_1, \ldots S'_n)$ satisfies the following– For all $(S_1, \ldots, S_n) \leftarrow f^{\alpha}_{\mathsf{ASh}}(v)$ and $(S'_1, \ldots, S'_n)$ such that $S'_i = S_i$ corresponding to at least $\alpha + 1$ parties P_i, it holds that $\Pr[v' \neq v] \leq \mathtt{negl}(\kappa)$ for a computational security parameter κ.*

iii. Round complexity. ARec^{α} *terminates in one round.*

Proof.

i. Privacy: It is easy to check from the description of $f^{\alpha}_{\mathsf{ASh}}$ that privacy follows directly from the fact that v is Shamir-shared with degree α.

ii. Correctness: Firstly, we note that since at least $\alpha+1$ parties P_i participate with $S'_i = S_i$ in $\mathsf{ARec}^{\alpha}(S'_1, \ldots S'_n)$, the Val set of each honest party comprises of at least $(\alpha + 1)$ correct shares sh_i. These shares suffice to uniquely reconstruct the α-shared secret v. We can thus conclude that an honest P_i would output $\perp$ only if the interpolation of the α-degree polynomial fails, which in turn occurs if there is an incorrect share, say sh'_j, such that j is added to Val. This would imply that a corrupt P_j broadcasts $\mathsf{sh}'_j \neq \mathsf{sh}_j$ and tag'_{ji} but satisfied the condition $\mathsf{Mac}_{k_{ji}}(\mathsf{sh}'_j) = \mathsf{tag}'_{ji}$, with respect to the MAC-key k_{ji} (present in S_i) available to honest P_i (not to P_j). However, security of MAC ensure that the above cannot happen except with negligible probability. This completes the proof of correctness.

iii. Round complexity. The proof is self-evident.

□

6.2 Upper bound for Robust MPC: The general case

In a setting where either at most $n-1$ passive corruption or at most $(\lceil \frac{n}{2} \rceil - 1)$ active corruption takes place, [IKK+11] presents a protocol relying on two types of MPC protocol. An actively-secure protocol against malicious majority is used to compute an authenticated-sharing of the output with threshold $(\lceil \frac{n}{2} \rceil - 1)$. When this protocol succeeds, the output is computed via reconstruction of the authenticated-sharing. On the other hand, a failure is tackled via running a robust honest-majority (majority of the parties are honest) actively-secure protocol, relying on the conclusion that the protocol is facing a malicious-minority. When n is odd, we need to tackle the exact corruption scenarios as that of the protocols of [IKK+11]. On the other hand when n is even, the extreme case for active corruption accommodates an additional passive corruption. Apart from hitting optimal round complexity, tackling the distinct boundary cases for odd and even n in a unified way brings challenge for our protocol.

We make the following effective changes to the approach of [IKK+11]. First, we invoke a 2-round actively-secure protocol π_{idua} with identifiable abort against malicious majority (can be instantiated with the protocol of [GS18], as shown by [CGZ20]) to compute $\lfloor \frac{n}{2} \rfloor$-authenticated sharing of the output. When we expel the identified corrupt parties in case of failure (which may occur in corruption scenario $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$), the remaining population always displays honest-majority, no matter whether n is odd or even (For instance, elimination of 1 corrupt party results in $t' \leq (t_p - 1) = \lfloor n/2 \rfloor - 1$ total corruptions among $n' = (n-1)$ remaining parties which satisfies $n' \geq 2t' + 1$.). The robust honest-majority protocol π_{god} is then invoked to compute the function f where the inputs of the identified parties are hard-coded to default values. The change in the degree of authenticated sharing ensures that an adversary choosing to corrupt in the boundary case of $\lceil \frac{n}{2} \rceil - 1$ active corruption and zero (when n is odd) or one (when n is even) purely passive corruption, cannot learn the output by itself collating the information it gathers during π_{idua}. Without the change, the adversary could ensure that π_{idua} leads to a failure for the honest parties and yet could learn outputs from both π_{idua} and π_{god} with different set of adversarial-inputs. Lastly, the function and input independence property of Round 1 of the 3-round honest-majority protocol of [GLS15, ACGJ18] allows us to superimpose this round with the run of π_{idua}. Both these instantiations of π_{god} are also equipped to tackle the probable change in population for the remaining two rounds (when identified corrupt parties are expelled) and the change in the function to be computed (with hard-coded default inputs for the identified corrupt parties).

We present our protocol $\pi_{\mathsf{god}}^{\mathsf{bou}}$ in the $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model in Fig. 9, where $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ (Fig. 8) denotes the ideal functionality (realized by π_{idua}) computing the authenticated sharing of the output securely with identifiable abort. Note that the hybrid model does not involve an ideal functionality corresponding to the 3-round subprotocol π_{god}, as Round 2 - 3 of the instance of π_{god} is executed only if needed, depending on the adversarial behaviour.

Assumption wise, $\pi_{\mathsf{god}}^{\mathsf{bou}}$ relies on 2-round maliciously-secure OT in the common random/reference string model (when π_{idua} is instantiated with function-delayed variant of the protocol of [GS18] satisfying identifiability) and Zaps and public-key encryption (when π_{god} is instantiated with the protocol of [ACGJ18]).

We state the formal theorem below.

Theorem 6. *Assuming the presence of a 3-round protocol π_{god} realizing $\mathcal{F}_{\mathsf{god}}$ in the presence of honest majority (with special property of Round 1 being function and input-independent), the 4-round MPC protocol $\pi_{\mathsf{god}}^{\mathsf{bou}}$ (Figure 9) in the $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model satisfies:*

- *Correctness: computes the correct output.*
- *Security: realizes $\mathcal{F}_{\mathsf{god}}$ (Fig 16) against boundary-admissible $\mathcal{A}$*

Proof. Correctness of $\pi_{\mathsf{god}}^{\mathsf{bou}}$ follows directly from that of $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$, π_{god} and $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ (Lemma 14). □

The security proof of $\pi_{\mathsf{god}}^{\mathsf{bou}}$ in the $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model appears in Appendix C.1. Standard composition theorems [Can00, Gol04] implies that $\pi_{\mathsf{god}}^{\mathsf{bou}}$ is secure when access to $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ is emulated using the 2-round function-delayed variant of the protocol of [GS18] satisfying identifiability.

We conclude this section with a simplification to $\pi_{\mathsf{god}}^{\mathsf{bou}}$ that can be adopted if additional access to PKI is assumed. In such a case, parallelizing Round 1 of π_{god}

Functionality $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$

Input: On message $(\mathtt{Input}, x_i)$ from a party P_i ($i \in [n]$), do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i sends no input, consider $x_i = (\mathtt{abort}, i)$. We require that if $x_i = (\mathtt{abort}, i)$, then the adversary corrupts P_i actively.

Output to adversary: If there exists a set $\mathcal{I}$ with $|\mathcal{I}| \geq 1$ such that $x_i = (\mathtt{abort}, i)$ for $P_i \in \mathcal{I}$, send $(\mathtt{Output}, (\bot, \mathcal{I}))$ to all the parties. Else, compute $y = f(x_1, \ldots, x_n)$ and $(S_1, \ldots S_n) = f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}(y)$ (Fig. 6). Send $(\mathtt{Output}, S_i)$ to P_i for every $P_i \in \mathcal{E}$.

Output to honest parties: Receive either $\mathtt{continue}$ or $(\mathtt{abort}, \mathcal{I})$ from adversary where $\mathcal{I}$ with $|\mathcal{I}| \geq 1$ is a subset of actively corrupt parties chosen by the adversary. In case of $\mathtt{continue}$, send $(\mathtt{Output}, S_i)$ to each honest P_i, whereas in case of $\mathtt{abort}$ send $(\mathtt{Output}, (\bot, \mathcal{I}))$ to all honest parties. We require that an adversary that corrupts no party actively sends $\mathtt{continue}$.

Fig. 8: Ideal Functionality $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$

Protocol $\pi_{\mathsf{god}}^{\mathsf{bou}}$

Inputs: Party P_i has x_i for $i \in [n]$

Model: $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model (Fig. 8)

Building Blocks: (a) 3-round honest-majority actively-secure robust protocol π_{god} (realizing functionality $\mathcal{F}_{\mathsf{god}}$, refer Fig 16) with additional property of Round 1 being function and input independent; (b) Protocol $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ for reconstructing an $\lfloor n/2 \rfloor$-authenticated-shared secret (Fig 7)

Output: $y = f(x_1, \ldots, x_n)$

Round 1–2: Each P_i interacts with $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ using input x_i to compute the function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f$ and obtain output $(S_i = \big(\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big), \mathcal{B})$, where $\mathcal{B}$ denotes the set of identified cheaters. Additionally, the parties run (input-independent and function-independent) Round 1 of π_{god}.

Round 3–4: If $S_i = \bot$, the parties in $\mathcal{P} \setminus \mathcal{B}$ run Round 2 and 3 of π_{god} computing $f^{\mathcal{B}}$ (f with the inputs of parties in $\mathcal{B}$ are hardcoded to default values) and output y as the outcome of π_{god}. Else, participate in $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ with input S_i and output the outcome of $\mathsf{ARec}^{\lfloor n/2 \rfloor}$.

Fig. 9: Robust MPC against boundary-admissible adversary

with Round 1 of π_{idua} (realizing $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$) can be avoided and the 2-round honest-majority protocol of [GLS15] achieving GOD assuming CRS and PKI setup can be used to instantiate π_{god} (which would be run in Rounds 3-4 of $\pi_{\mathsf{god}}^{\mathsf{bou}}$). Both our 4-round constructions with CRS (Figure 9) and its simplified variant with CRS and PKI are tight upper bounds, in light of the impossibility of Section 5.1 that holds in the presence of CRS and PKI.

6.3 Upper bound for Robust MPC: The single corruption case

Building upon the ideas of Section 6.2 and Section 4.3, a 3-round robust MPC $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ against the special-case boundary-admissible adversary can be constructed as follows. Similar to $\pi_{\mathsf{god}}^{\mathsf{bou}}$, Round 1 and 2 involve running protocol π_{idua} realizing $\lfloor n/2 \rfloor$-authenticated secret-sharing of the function output. When π_{idua} does not result in abort, $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ proceeds to reconstruction of output; identical to $\pi_{\mathsf{god}}^{\mathsf{bou}}$ and thereby terminating in 3 rounds. However, when π_{idua} results in output $\perp$, we exploit the advantage of at most one malicious corruption by noting that once the single actively-corrupt party is expelled, the parties involved thereafter comprise only of the honest and purely passive parties. We adopt the idea of Section 4.3 and re-run Round 2 of π_{idua} among the remaining parties to compute the function output directly, with input of the expelled party substituted with default input. This step demands the function-delayed property of π_{idua} i.e. Round 1 is independent of the function to be computed and the number of parties. In order to accommodate this re-run, two instances of Round 1 of π_{idua} are run in Round 1 of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$. It is easy to see that robustness is ensured as π_{idua} is robust in the absence of actively-corrupt parties. Lastly, we point that similar to Section 4.3, we use the modified variant of the 2-round protocol of [GS18] to instantiate π_{idua} that is function-delayed and achieves identifiability.

The formal description of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ appears in Fig 10. $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ is not analyzed in the hybrid model as it is not possible to substitute the second instance of π_{idua} with its corresponding ideal functionality in the hybrid model. This is because only Round 1 of the second instance of π_{idua} is always executed, but its Round 2 is run only if needed, depending on adversarial behaviour.

This upper bound $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ is tight, following the impossibility of 2-round fair MPC (that holds for single malicious corruption) proven in Section 5.2 as GOD implies fairness. Assumption wise, $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ relies on 2-round maliciously-secure OT in the common random/reference string model (when π_{idua} is instantiated with function-delayed variant of the protocol of [GS18] satisfying identifiability).

We state the formal theorem below.

Theorem 7. *Assuming the presence of a 2-round protocol π_{idua} realizing functionality $\mathcal{F}_{\mathsf{idua}}$ (Fig 14) against malicious majority and having function-delayed property, the 3-round MPC protocol $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ (Figure 10) satisfies:*

- *Correctness: computes the correct output.*
- *Security: realizes $\mathcal{F}_{\mathsf{god}}$ (Fig 16) against special-case boundary-admissible $\mathcal{A}$ with corruption parameters either $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ or $(t_a, t_p) = (0, n-1)$.*

Proof. Correctness of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ follows directly from correctness of π_{idua}, and correctness of $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ (Lemma 14). □

The security proof is deferred to Appendix C.2.

6.4 Upper bound for Fair MPC

The 4-round robust protocol $\pi_{\mathsf{god}}^{\mathsf{bou}}$ (Section 6.2) can be simplified as follows to yield a 3-round fair MPC protocol $\pi_{\mathsf{fair}}^{\mathsf{bou}}$. Similar to $\pi_{\mathsf{god}}^{\mathsf{bou}}$, Round 1 and 2 involve

Protocol $\pi^{\mathsf{bou},1}_{\mathsf{god}}$

Inputs: Party P_i has x_i for $i \in [n]$

Building Blocks: (a) 2-round protocol π_{idua} achieving identifiable abort against malicious majority (realizing functionality $\mathcal{F}_{\mathsf{idua}}$, refer Fig. 14) and having function-delayed property; (b) Protocol $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ for reconstructing an $\lfloor n/2 \rfloor$-authenticated-shared secret (Fig. 7); (c) Function $f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}}$ (Fig. 6).

Output: $y = f(x_1, \ldots, x_n)$

Round 1: P_i does the following: Run 2 instances of Round 1 of π_{idua}, each using input x_i and independent randomness. Note that this round is independent of the function to be computed and the number of parties.

Round 2: P_i does the following: Run Round 2 of π_{idua} (based on first instance of Round 1 of π_{idua}) among $\mathcal{P}$ computing the function $f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}} \diamond f$ using input x_i to obtain output $(S_i = \big(\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big), \mathcal{B})$, where $\mathcal{B}$ denotes the set of identified cheaters.

Round 3: If $S_i = \perp$, the parties in $\mathcal{P} \setminus \mathcal{B}$ run Round 2 of π_{idua} (based on second instance of Round 1 of π_{idua}) computing $f^{\mathcal{B}}$ (where the inputs of the party in $\mathcal{B}$ is hard-coded to default value) and output y as the outcome of this (second) instance of π_{idua}. Else, participate in $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ with input S_i and output the outcome of $\mathsf{ARec}^{\lfloor n/2 \rfloor}$.

Fig. 10: Robust MPC against special-case boundary-admissible adversary

execution of π_{ua} (instantiated by [GS18, BL18] in the CRS model) achieving unanimous abort against malicious-majority (identifiability is not needed) in order to compute $\lfloor n/2 \rfloor$-authenticated sharing of the output. If π_{ua} does not result in abort, the honest parties proceed to reconstruction of output in Round 3. Else, the honest parties simply output $\perp$. It is easy to check that fairness is preserved due to privacy of $\lfloor n/2 \rfloor$-authenticated secret-sharing (Lemma 14).

Protocol $\pi^{\mathsf{bou}}_{\mathsf{fair}}$ in the $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{ua}}$-hybrid model appears in Fig 12, where $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{ua}}$ (Fig. 11) denotes the ideal functionality computing the authenticated sharing of the output securely with abort. When $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{ua}}$ is realized using the 2-round protocols of [GS18, BL18], $\pi^{\mathsf{bou}}_{\mathsf{fair}}$ is round-optimal, in view of the lower bound of Section 5.2 and relies on 2-round maliciously-secure OT in the common random/reference string model.

We state the formal theorem below.

Theorem 8. *The 3-round MPC protocol $\pi^{\mathsf{bou}}_{\mathsf{fair}}$ (Figure 12) in the $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{ua}}$-hybrid model satisfies:*

- *Correctness: computes the correct output.*
- *Security: realizes against (t_a, t_p) boundary-admissible $\mathcal{A}$ (1) $\mathcal{F}_{\mathsf{fair}}$ (Fig. 15) when $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ (2) $\mathcal{F}_{\mathsf{god}}$ (Fig. 16) when $(t_a, t_p) = (0, n-1)$.*

Proof. Correctness of $\pi^{\mathsf{bou}}_{\mathsf{fair}}$ follows directly from correctness of $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{ua}}$ and the correctness of $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ (Lemma 14). □

Functionality $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$

Input: On message (Input, x_i) from a party P_i $(i \in [n])$, do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i does not send an input, consider $x_i = \mathtt{abort}$. We require that when $x_i = \mathtt{abort}$, the adversary corrupts P_i actively.

Output to adversary: If there exists $i \in [n]$ such that $x_i = \mathtt{abort}$, send $(\mathtt{Output}, \perp)$ to all the parties. Else, compute $y = f(x_1, \ldots, x_n)$ and $(S_1, \ldots, S_n) = f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}(y)$. Send $(\mathtt{Output}, S_i)$ to P_i for every $P_i \in \mathcal{E}$.

Output to honest parties: Receive either $\mathtt{continue}$ or $\mathtt{abort}$ from adversary. In case of $\mathtt{continue}$, send $(\mathtt{Output}, S_i)$ to each honest P_i, whereas in case of $\mathtt{abort}$ send $(\mathtt{Output}, \perp)$ to all honest parties. We require that an adversary that corrupts no party actively sends $\mathtt{continue}$.

Fig. 11: Ideal Functionality $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$

Protocol $\pi_{\mathsf{fair}}^{\mathsf{bou}}$

Inputs: Party P_i has x_i for $i \in [n]$

Model: $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model (Fig. 11)

Building Blocks: Protocol $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ for reconstructing an $\lfloor n/2 \rfloor$-authenticated-shared secret (Fig. 7)

Output: $y = f(x_1, \ldots, x_n)$ or $\perp$.

Round 1–2: P_i interacts with $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ with input x_i to compute the function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f$ and obtains S_i as output.

Round 3: If $S_i = \perp$, the parties output $\perp$. Else, participate in $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ with input $S_i = \big(\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}\big)$ and output the outcome of $\mathsf{ARec}^{\lfloor n/2 \rfloor}$.

Fig. 12: Fair MPC against boundary-admissible adversary

The security proof of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$ in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model appears in Appendix C.3. Standard composition theorems [Can00, Gol04] implies that $\pi_{\mathsf{fair}}^{\mathsf{bou}}$ is secure when access to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ is emulated using the 2-round protocols of [GS18, BL18].

References

ABT19. Benny Applebaum, Zvika Brakerski, and Rotem Tsabary. Degree 2 is complete for the round-complexity of malicious MPC. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, pages 504–531, 2019.

ACGJ18. Prabhanjan Ananth, Arka Rai Choudhuri, Aarushi Goel, and Abhishek Jain. Round-optimal secure multiparty computation with honest majority. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 395–424, 2018.

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 *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, pages 231–250, 2008.

BJMS18. Saikrishna Badrinarayanan, Aayush Jain, Nathan Manohar, and Amit Sahai. Secure MPC: laziness leads to GOD. *IACR Cryptology ePrint Archive*, 2018:580, 2018.

BKP11. Michael Backes, Aniket Kate, and Arpita Patra. Computational verifiable secret sharing revisited. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 590–609, 2011.

BL18. Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 500–532, 2018.

Can00. R. Canetti. Security and Composition of Multiparty Cryptographic Protocols. *J. Cryptology*, 13(1):143–202, 2000.

CDG87. David Chaum, Ivan Damgård, and Jeroen Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 87–119, 1987.

CGZ20. Ran Cohen, Juan A. Garay, and Vassilis Zikas. Broadcast-optimal two-round MPC. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, pages 828–858, 2020.

Cha89. David Chaum. The spymasters double-agent problem: Multiparty computations secure unconditionally from minorities and cryptographically from majorities. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 591–602, 1989.

CL14. Ran Cohen and Yehuda Lindell. Fairness versus guaranteed output delivery in secure multiparty computation. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Applica-*

tion of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II, pages 466–485, 2014.

Cle86. Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 364–369, 1986.

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

FHHW03. Matthias Fitzi, Martin Hirt, Thomas Holenstein, and Jürg Wullschleger. Two-threshold broadcast and detectable multi-party computation. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 51–67, 2003.

FHM98. Matthias Fitzi, Martin Hirt, and Ueli M. Maurer. Trading correctness for privacy in unconditional multi-party computation (extended abstract). In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 121–136, 1998.

FHM99. Matthias Fitzi, Martin Hirt, and Ueli M. Maurer. General adversaries in unconditional multi-party computation. In *Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Singapore, November 14-18, 1999, Proceedings*, pages 232–246, 1999.

FHW04. Matthias Fitzi, Thomas Holenstein, and Jürg Wullschleger. Multi-party computation with hybrid security. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 419–438, 2004.

GIKR02. Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. On 2-round secure multiparty computation. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 178–193, 2002.

GLS15. S. Dov Gordon, Feng-Hao Liu, and Elaine Shi. Constant-round MPC with fairness and guarantee of output delivery. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 63–82, 2015.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987.

Gol01. Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques.* Cambridge University Press, 2001.

Gol04. Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications.* Cambridge University Press, 2004.

GS18. Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 468–499, 2018.

HLM13. Martin Hirt, Christoph Lucas, and Ueli Maurer. A dynamic tradeoff between active and passive corruptions in secure multi-party computation. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 203–219, 2013.

HLMR11. Martin Hirt, Christoph Lucas, Ueli Maurer, and Dominik Raub. Graceful degradation in multi-party computation (extended abstract). In *Information Theoretic Security - 5th International Conference, ICITS 2011, Amsterdam, The Netherlands, May 21-24, 2011. Proceedings*, pages 163–180, 2011.

HLP11. Shai Halevi, Yehuda Lindell, and Benny Pinkas. Secure computation on the web: Computing without simultaneous interaction. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 132–150, 2011.

HM01. Martin Hirt and Ueli M. Maurer. Robustness for free in unconditional multi-party computation. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 101–118, 2001.

HMP00. Martin Hirt, Ueli M. Maurer, and Bartosz Przydatek. Efficient secure multi-party computation. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, pages 143–161, 2000.

HMZ08. Martin Hirt, Ueli M. Maurer, and Vassilis Zikas. MPC vs. SFE : Unconditional and computational security. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 1–18, 2008.

IKK$^+$11. Yuval Ishai, Jonathan Katz, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. On achieving the "best of both worlds" in secure multiparty computation. *SIAM J. Comput.*, 40(1):122–141, 2011.

IKKP15. Yuval Ishai, Ranjit Kumaresan, Eyal Kushilevitz, and Anat Paskin-Cherniavsky. Secure computation with minimal interaction, revisited. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 359–378, 2015.

IKLP06. Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. On combining privacy with guaranteed output delivery in secure multiparty computation. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 483–500, 2006.

IKP$^+$16. Yuval Ishai, Eyal Kushilevitz, Manoj Prabhakaran, Amit Sahai, and Ching-Hua Yu. Secure protocol transformations. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 430–458, 2016.

Kat07. Jonathan Katz. On achieving the "best of both worlds" in secure multiparty computation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 11–20, 2007.

Lin17. Yehuda Lindell. How to simulate it - A tutorial on the simulation proof technique. In *Tutorials on the Foundations of Cryptography.*, pages 277–346. 2017.

LRM10. Christoph Lucas, Dominik Raub, and Ueli M. Maurer. Hybrid-secure MPC: trading information-theoretic robustness for computational privacy. In *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*, pages 219–228, 2010.

PCRR09. Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 487–504, 2009.

Ped91. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 129–140, 1991.

PR18a. Arpita Patra and Divya Ravi. On the exact round complexity of secure three-party computation. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 425–458, 2018.

PR18b. Arpita Patra and Divya Ravi. On the exact round complexity of secure three-party computation. *IACR Cryptology ePrint Archive*, 2018:481, 2018.

RB89. Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85, 1989.

Sha79. Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

Yao82. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.

A Security Model

We prove the security of our protocols based on the standard real/ideal world paradigm. Essentially, the security of a protocol is analyzed by comparing what an adversary can do in the real execution of the protocol to what it can do in an ideal execution, that is considered secure by definition (in the presence of an incorruptible trusted party). In an ideal execution, each party sends its input to the trusted party over a perfectly secure channel, the trusted party computes the function based on these inputs and sends to each party its respective output. Informally, a protocol is secure if whatever an adversary can do in the real protocol (where no trusted party exists) can be done in the above described ideal computation. We refer to [Can00, Gol01, Lin17, CL14] for further details regarding the security model.

The “ideal" world execution involves n parties $\{P_1 \ldots, P_n\}$, an ideal adversary $\mathcal{S}$ who may corrupt some of the parties, and a functionality $\mathcal{F}$. The “real" world execution involves the PPT parties $\{P_1 \ldots, P_n\}$, and a real world PPT adversary $\mathcal{A}$ who may corrupt some of the parties. We let $\textsc{ideal}_{\mathcal{F},\mathcal{S}}(1^\kappa, z)$ denote the output pair of the honest parties and the ideal-world PPT adversary $\mathcal{S}$ from the ideal execution with respect to the security parameter 1^κ and auxiliary input z. Similarly, let $\textsc{real}_{\Pi,\mathcal{A}}(1^\kappa, z)$ denote the output pair of the honest parties and the adversary $\mathcal{A}$ from the real execution with respect to the security parameter 1^κ and auxiliary input z.

Definition 5. *For $n \in \mathbb{N}$, let $\mathcal{F}$ be a functionality and let Π be a n-party protocol. We say that Π securely realizes $\mathcal{F}$ if for every PPT real world adversary $\mathcal{A}$, there exists a PPT ideal world adversary $\mathcal{S}$, corrupting the same parties, such that the following two distributions are computationally indistinguishable: $\textsc{ideal}_{\mathcal{F},\mathcal{S}} \overset{c}{\approx} \textsc{real}_{\Pi,\mathcal{A}}$.*

Target Functionalities. Taking motivation from [CL14, GLS15], we define ideal functionalities $\mathcal{F}_{\mathsf{ua}}$, $\mathcal{F}_{\mathsf{idua}}, \mathcal{F}_{\mathsf{fair}}, \mathcal{F}_{\mathsf{god}}$ in Figures 13, 14, 15, 16 for secure MPC of a function f with unanimous abort, identifiable abort, fairness and guaranteed output delivery respectively.

A.1 Equivocal Non-interactive Commitment Schemes (eNICOM)

We present the formal definition and properties of Equivocal Non-interactive Commitment Schemes (eNICOM) in this section. This primitive is used in our upper bound constructions against dynamic-admissible adversary.

An equivocal non-interactive commitment scheme (eNICOM) consists of algorithms $(\mathsf{eGen}, \mathsf{eCom}, \mathsf{eOpen}, \mathsf{Equiv})$ defined as follows.

- $\mathsf{eGen}(1^\kappa)$ returns a public parameter and a corresponding trapdoor (epp, t), where epp is used by both eCom and eOpen. The trapdoor t is used for equivocation.

$\mathcal{F}_{\mathsf{ua}}$

Input: On message (Input, x_i) from a party P_i $(i \in [n])$, do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i sends no input, consider $x_i = \texttt{abort}$. We require that if $x_i = \texttt{abort}$, then the adversary corrupts P_i actively.

Output to adversary: If there exists $i \in [n]$ such that $x_i = \texttt{abort}$, send $(\texttt{Output}, \bot)$ to all the parties. Else, send $(\texttt{Output}, y)$ to the adversary, where $y = f(x_1, \ldots, x_n)$.

Output to honest parties: Receive either $\texttt{continue}$ or $\texttt{abort}$ from adversary. In case of $\texttt{continue}$, send $(\texttt{Output}, y)$ to honest parties, whereas in case of $\texttt{abort}$, send them $(\texttt{Output}, \bot)$. We require that an adversary that corrupts no party actively sends $\texttt{continue}$.

Fig. 13: Ideal Functionality for unanimous abort

$\mathcal{F}_{\mathsf{idua}}$

Input: On message (Input, x_i) from a party P_i $(i \in [n])$, do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i sends no input, consider $x_i = (\texttt{abort}, i)$. We require that if $x_i = (\texttt{abort}, i)$, then the adversary corrupts P_i actively.

Output to adversary: If there exists a set $\mathcal{I}$ with $|\mathcal{I}| \geq 1$ such that $x_i = (\texttt{abort}, i)$ for $P_i \in \mathcal{I}$, send $(\texttt{Output}, (\bot, \mathcal{I}))$ to all the parties. Else, send $(\texttt{Output}, y)$ to the adversary, where $y = f(x_1, \ldots, x_n)$.

Output to honest parties: Receive either $\texttt{continue}$ or $(\texttt{abort}, \mathcal{I})$ from adversary where $\mathcal{I}$ with $|\mathcal{I}| \geq 1$ is a subset of actively corrupt parties chosen by the adversary. In case of $\texttt{continue}$, send $(\texttt{Output}, y)$ to honest parties, whereas in case of $\texttt{abort}$ send $(\texttt{Output}, (\bot, \mathcal{I}))$ to all honest parties. We require that an adversary that corrupts no party actively sends $\texttt{continue}$.

Fig. 14: Ideal Functionality for identifiable abort

$\mathcal{F}_{\mathsf{fair}}$

Input: On message (Input, x_i) from a party P_i $(i \in [n])$, do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i sends no input, consider $x_i = \texttt{abort}$. We require that if $x_i = \texttt{abort}$, then the adversary corrupts P_i actively.

Output: If there exists $i \in [n]$ such that $x_i = \texttt{abort}$, send $(\texttt{Output}, \bot)$ to all the parties. Else, send $(\texttt{Output}, y)$ to party P_i for every $i \in [n]$, where $y = f(x_1, \ldots, x_n)$.

Fig. 15: Ideal Functionality for fairness

- $\mathsf{eCom}(\mathsf{epp}, x; r)$ returns a commitment c and corresponding opening information o (where inputs are the common parameter epp, message x and random coins r).

$\mathcal{F}_{\mathsf{god}}$

Input: On message (Input, x_i) from a party P_i $(i \in [n])$, do the following: if such a message was received from P_i earlier, then ignore. Otherwise record it internally. If x_i is outside of the domain for P_i or P_i sends no input, set x_i to be some predetermined default value.

Output: Compute $y = f(x_1, \ldots, x_n)$ and send $(\mathtt{Output}, y)$ to party P_i for every $i \in [n]$.

Fig. 16: Ideal Functionality for guaranteed output delivery

– $\mathsf{eOpen}(\mathsf{epp}, c, o)$ returns the message x.
– $\mathsf{Equiv}(c, o', x, t)$ is invoked on commitment c and its corresponding opening o', given message x and the trapdoor t and returns o such that $x \leftarrow \mathsf{eOpen}(\mathsf{epp}, c, o)$.

Informally, the algorithms should satisfy correctness, binding (i.e. it must be hard for an adversary to come up with two different openings of any c with respect to uniformly chosen epp) and hiding (a commitment must not leak information about the underlying message) properties. The hiding property of eNICOM is slightly changed compared to that of standard non-interactive commitments taking the equivocation property into account. This new definition implies the usual hiding definition.

Properties.

– *Correctness:* For all $(\mathsf{epp}, t) \leftarrow \mathsf{eGen}(1^\kappa)$, $x \in \mathcal{M}$ and $r \in \mathcal{R}$, if $(c, o) \leftarrow \mathsf{eCom}(x; r)$ then $\mathsf{eOpen}(c, o) = x$.
– *Binding:* For all $(\mathsf{epp}, t) \leftarrow \mathsf{eGen}(1^\kappa)$ and for all PPT adversaries $\mathcal{A}$, it is with negligible probability that $\mathcal{A}(\mathsf{epp})$ outputs (c, o, o') such that $\mathsf{eOpen}(c, o) \neq \mathsf{eOpen}(c, o')$ and $\perp \notin \{\mathsf{eOpen}(c, o), \mathsf{eOpen}(c, o')\}$
– *Hiding:* For all $(\mathsf{epp}, t) \leftarrow \mathsf{eGen}(1^\kappa)$ and for all PPT adversaries $\mathcal{A}$, and all $x, x' \in \mathcal{M}$, the following difference is negligible:

$$\left|\mathsf{Pr}_{(c,o)\leftarrow\mathsf{eCom}(x)}[\mathcal{A}(c,o)=1] - \mathsf{Pr}_{(c,o')\leftarrow\mathsf{eCom}(x'),o\leftarrow\mathsf{Equiv}(c,o',x,t)}[\mathcal{A}(c,o)=1]\right|$$

Instantiation. We present the instantiation based on Pedersen commitment scheme [Ped91].

Theorem 9. *Let p, q denote large primes such that q divides $(p-1)$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q and g is a generator of G_q. Consider the following algorithms:*

- $\mathsf{eGen}(1^\kappa)$*: set* $(\mathsf{epp}, t) = ((g, h), \alpha)$ *where* $\alpha \in \mathbb{Z}_q$*;* $h = g^\alpha$
- $\mathsf{eCom}(\mathsf{epp} = (g, h), x; r)$*: set* $c = g^x h^r$*; set* $o = (r, x)$*.*
- $\mathsf{eOpen}(\mathsf{epp} = (g, h), c, o = (r, x))$*: return* x *if* $c = g^x h^r$*; otherwise return* $\perp$*.*
- $\mathsf{Equiv}((c = \mathsf{eCom}(x'; r')), (x', r'), x, t)$*: return* $o = (r, x)$ *where* $r = r' + \frac{x' - x}{t}$

Assume that the discrete logarithm problem in G_q is hard. Then $(\mathsf{eGen}, \mathsf{eCom}, \mathsf{eOpen}, \mathsf{Equiv})$ *is an equivocal commitment scheme.*

Proof. It is easy to check that correctness holds. We prove the remaining properties below.

- *Binding.* Assume towards a contradiction that there exist a PPT adversary $\mathcal{A}$ such that $\mathcal{A}(\mathsf{epp})$ outputs (c, o, o') such that $\mathsf{eOpen}(c, o) \neq \mathsf{eOpen}(c, o')$ and $\perp \notin \{\mathsf{eOpen}(c, o), \mathsf{eOpen}(c, o')\}$ with non-negligible probability. We show that $\mathcal{A}$ can be used to construct an adversary $\mathcal{A}'$ to find the discrete logarithm. $\mathcal{A}'$ forwards its input (g, h) to $\mathcal{A}$ as epp who returns (c, o, o'). Let $x = \mathsf{eOpen}(c, o)$ and $x' = \mathsf{eOpen}(c, o)$, where $o = (r, x)$ and $o' = (r', x')$. If $\mathcal{A}'$ succeeded in breaking the binding property, it must hold that $c = g^x h^r = g^{x'} h^{r'}$. Therefore, $\mathcal{A}'$ can compute the discrete logarithm of h as $\log_g h = \frac{x'-x}{r-r'}$. We can thus conclude that $\mathcal{A}'$ succeeds in computing the discrete logarithm, provided $\mathcal{A}$ succeeds in breaking the binding property, which occurs with non-negligible probability. This is a contradiction to our assumption that the discrete logarithm problem is hard.

- *Hiding.* The commitment scheme is perfectly hiding i.e. $c = g^x h^r$ reveals no information about x. This holds, because h^r has a uniform distribution over G_q, independently of the choice of x (as r is chosen uniformly at random from $\mathbb{Z}_q$).

□

B Proofs for Upper Bounds for Dynamic Corruption

B.1 Security Proof of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ (Theorem 2)

Proof. We analyze the protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ in a $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model where the parties have access to a trusted party $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ (Fig. 3). Standard composition theorems [Can00, Gol04] implies that $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ is secure when access to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ is emulated using 2-round protocols of [GS18, BL18]. Let $\mathcal{A}$ be a dynamic adversary with threshold (t_a, t_p) that controls t_p parties passively and upto t_a among them actively in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$.

We prove Theorem 2 by describing a simulator for each admissible corruption scenario (t_a, t_p) of $\mathcal{A}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{fair}}$ (refer Figure 15) computing f whose behaviour simulates the behaviour of $\mathcal{A}$. While the Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$ corresponding to the case of $t_a = 0$ appears in Fig. 17, the Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ (parameterized by t_a, where $t_a \geq 1$) in Fig. 18 describes the simulation steps corresponding to all corruption scenarios where $t_a \geq 1$.

In order to complete the proof, we argue how each of them maintain that the view of $\mathcal{A}$ is the ideal world is indistinguishable from its view in $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ (hybrid-world).

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$

Let H and Corr denote the set of *indices* of honest parties (in $\mathcal{P} \setminus \mathcal{E}$) and parties in $\mathcal{E}$ respectively. $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$ does the following:

- **Interaction with $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$:** Receive $(\mathsf{Input}, \{x_j\}_{j \in \mathsf{Corr}})$ sent by $\mathcal{A}$ to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$.
- **Output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ to $\mathcal{A}$:** Invoke $\mathcal{F}_{\mathsf{fair}}$ on behalf of $\mathcal{A}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ to receive an output value y in return. Compute $(L_1, \ldots L_n) = f_{\mathsf{LSh}}^{n-2,\lfloor \frac{n}{2} \rfloor}(y)$ (Fig. 1) and return $\{L_j\}_{j \in \mathsf{Corr}}$ to $\mathcal{A}$ as output from $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$. Receive `continue` sent by $\mathcal{A}$ to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$.

  Note: Recall that in Round r ($r \in [3, \lceil n/2 \rceil + 1]$), summand s_{n-r+1} is attempted to be reconstructed (in Round $r-2$ of $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$).
- **Round 3 to Round $\lceil n/2 \rceil + 1$:** $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$ does the following in Round r', where $r' = [3, \lceil n/2 \rceil + 1]$: Let $i = n - r' + 1$. Send $(s_{il}, o_{il}) \in L_l$ on behalf of honest $P_l \in \mathsf{H}$.

Fig. 17: Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},0}$ corresponding to $t_a = 0$. It is straightforward to see that the view of $\mathcal{A}$ is identical in the ideal and hybrid-world.

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ corresponding to $t_a \geq 1$. The simulation is divided into 4 parts: Computation of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ (comprising Rounds 1-2 when realized by 2-round protocols of [GS18, BL18]), Rounds 3 to $n - t_p$, Round $n - t_p + 1$ and finally Rounds $n - t_p + 1$ to Round $\lceil n/2 \rceil + 1$.

Computation of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$. The only difference between the ideal and hybrid-world is that the output obtained by adversary from $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ in the ideal world when $t_a \geq 1$, comprises of commitments on dummy values corresponding to shares of honest parties. This is in contrast to the hybrid world where the commitments are computed on the levelled-shares of the function output. We note that when $t_a \geq 1, t_p \leq n - 2$ holds; thereby $\mathcal{A}$ has access to at most $n - 2$ shares of the summand s_{n-2} (which is shared with threshold $n - 2$). Indistiguishability of the view of $\mathcal{A}$ in the ideal world and hybrid world follows directly from the property of Shamir-Sharing and hiding property of eNICOM.

Rounds 3 to Round $\mathbf{n - t_p}$. The only difference in the ideal and the hybrid-model is the following: In the hybrid-model, the share of a party P_j, say s_{ij} ($i = [n-2, t_p + 1]$) is discarded during $\mathsf{LRec}^{n-2,\lfloor n/2 \rfloor}()$ if the corresponding commitment c_{ij} (output from $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$) does not open successfully using the given opening o'_{ij} obtained from P_j. However, in the ideal world, the share of P_j is discarded if P_j does not send (s_{ij}, o_{ij}), same as output from $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$. It follows from the binding property of the equivocal commitment eNICOM that P_j will not be able to send $(s'_{ij}, o'_{ij}) \neq (s_{ij}, o_{ij})$ such that $\mathsf{eOpen}(\mathsf{epp}, c_{ij}, o'_{ij}) = s'_{ij}$, except with negligible probability. Thus, indistinguishability holds.

Round $\mathbf{n - t_p + 1}$. Since this constitutes the crux of the simulation, we first briefly describe the logic behind this simulation step - Note that if reconstructions of summands upto s_{t_p+1} were successful, in the hybrid-world, $\mathcal{A}$

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$

Let H and Corr denote the set of *indices* of honest parties (in $\mathcal{P} \setminus \mathcal{E}$) and parties in $\mathcal{E}$ respectively. $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ (where $t_a \geq 1$) does the following.

- **Interaction with $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$:** Receive $(\mathsf{Input}, \{x_j\}_{j \in \mathsf{Corr}})$ sent by $\mathcal{A}$ to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$.
- **Output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ to $\mathcal{A}$:**
  - If for any P_j, x_j is outside of domain of input, send $\perp$ as output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ to $\mathcal{A}$ and send $\perp$ as input to $\mathcal{F}_{\mathsf{fair}}$ on behalf of $\mathcal{A}$. This completes the simulation.
  - Else, let $\alpha' = n - 2$ and $\beta' = \lfloor n/2 \rfloor$. For $j \in \mathsf{Corr}$, return $L_j = \big(\{s_{ij}, o_{ij}\}_{i \in [\alpha', \beta']}, \{c_{il}\}_{i \in [\alpha', \beta'], l \in [n]}\big)$ where s_{ij} are randomly chosen, $(c_{ij}, o_{ij}) \leftarrow \mathsf{eCom}(s_{ij}; r_{ij})$ (with trapdoor t) computed as per protocol specifications and $\{c_{il}\}_{i \in [\alpha', \beta'], l \in \mathsf{H}}$ are computed as commitments on dummy values, say involving $\{s'_{il}, o'_{il}\}_{i \in [\alpha', \beta'], l \in \mathsf{H}}$.
- If $\mathcal{A}$ invokes $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ with `abort`, invoke $\mathcal{F}_{\mathsf{fair}}$ with input $\perp$ on behalf of $\mathcal{A}$; which completes the simulation. Else, continue to execute the remaining steps.

Note: Recall that in Round r ($r \in [3, \lceil n/2 \rceil + 1]$), summand s_{n-r+1} is attempted to be reconstructed (in Round $r - 2$ of $\mathsf{LRec}^{n-2, \lfloor \frac{n}{2} \rfloor}$).

- **Round 3 to Round $(\mathbf{n} - \mathbf{t_p})$:** $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ does the following in Round r', where $r' = [3, n - t_p]$
  - Let $i = n - r' + 1$. Send $\{s'_{il}, o'_{il}\}_{l \in \mathsf{H}}$ on behalf of honest parties and receive $\{s'_{ij}, o'_{ij}\}_{j \in \mathsf{Corr}}$ from $\mathcal{A}$.
  - Initialize $\mathsf{Val}_i = \mathcal{P} \setminus \mathcal{E}$. Add $P_j \in \mathcal{E}$ to Val_i if P_j sends $(s'_{ij}, o'_{ij}) = (s_{ij}, o_{ij})$ (consistent with L_j returned as output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ to P_j). If $|\mathsf{Val}_i| < i + 1$, then abort and invoke $\mathcal{F}_{\mathsf{fair}}$ with input $\perp$ on behalf of $\mathcal{A}$; thereby completing simulation. Else, continue to $r' = r' + 1$.
- **Round $(\mathbf{n} - \mathbf{t_p} + \mathbf{1})$:** This round involves reconstruction of summand s_{t_p}. $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ does the following:
  - Invoke $\mathcal{F}_{\mathsf{fair}}$ on behalf of $\mathcal{A}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ to receive output y.
  - Note that reconstruction of summands $s_{t_p+1}, \ldots, s_{n-2}$ has been completed and the summands s_i, where $i \in [\lfloor n/2 \rfloor, \ldots, t_p - 1]$ is already fully determined by $\{s_{ij}\}_{j \in \mathsf{Corr}}$ returned as output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$ to $\mathcal{A}$. Compute $s_{t_p} = y - \sum_{i=\lfloor n/2 \rfloor}^{t_p - 1} s_i - \sum_{i=t_p+1}^{n-2} s_i$.
  - Let $\mu = t_p$. Interpolate a μ-degree polynomial $g_\mu(x)$ satisfying $g_\mu(0) = s_\mu$ and $g_\mu(j) = s_{\mu j}$ for $j \in \mathsf{Corr}$. Let $s_{\mu l} = g_\mu(l)$ for $l \in \mathsf{H}$. Compute $o_{\mu l} \leftarrow \mathsf{Equiv}(c_{\mu l}, (s'_{\mu l}, o'_{\mu l}), s_{\mu l}, t)$ (Section A.1). Broadcast $(s_{\mu l}, o_{\mu l})$ on behalf of P_l, $l \in \mathsf{H}$.
- **Round $(\mathbf{n} - \mathbf{t_p} + \mathbf{2})$ to Round $(\lceil \mathbf{n/2} \rceil + \mathbf{1})$:** In Round r' ($r' \in [n - t_p + 2, \lceil n/2 \rceil + 1]$), broadcast (s'_{il}, o'_{il}) on behalf of P_l, $l \in \mathsf{H}$; where $i = n - r' + 1$.

Fig. 18: Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$

can deduce the output in Round $n - t_p + 1$ involving reconstruction of s_{t_p} (Summands $s_{\lfloor n/2 \rfloor}, \ldots, s_{t_p - 1}$ are already fully determined by output of $\mathcal{A}$ received from $\mathcal{F}_{\mathsf{ua}}^{\mathsf{LSh}}$). To maintain indistiguishability between the ideal and hybrid-world, $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ invokes $\mathcal{F}_{\mathsf{fair}}$ to obtain output y and sets s_{t_p} accordingly so that $\sum_{i=\lfloor n/2 \rfloor}^{n-2} s_i = y$. To argue indistinguishability, we note that the only difference between the ideal and hybrid-world is the following: In the hybrid world, for $i = t_p$, the commitments $\{c_{il}\}_{l \in \mathsf{H}}$ correspond to $\{s_{il}, o_{il}\}$ computed as per output y. However in the ideal world, $\{c_{il}\}_{l \in \mathsf{H}}$ were commitments on dummy values that were later equivocated to appropriate values of shares of honest parties as per the computed s_{t_p} (set such that the summands add upto y). Indistinguishability follows from the properties of equivocal commitment schemes (Section A.1).

Round $n - t_p + 2$ to Round $(\lceil n/2 \rceil + 1)$. It is easy to check that the view of $\mathcal{A}$ is identical in the ideal and hybrid-world.

This completes the proof of Theorem 2. □

B.2 Security Proof of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ (Theorem 3)

Proof. Let $\mathcal{A}$ be a dynamic-admissible adversary with threshold (t_a, t_p) that controls t_p parties passively and upto t_a among them actively during an execution of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$.

We prove Theorem 3 by describing a simulator for each admissible corruption scenario (t_a, t_p) of $\mathcal{A}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{god}}$ (refer Figure 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$.

While the Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$ corresponding to the case of $t_a = 0$ appears in Fig. 19, the Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ (parameterized by t_a, where $t_a \geq 1$) in Fig. 20-21 describes the simulation steps corresponding to all corruption scenarios where $t_a \geq 1$. These simulators invoke the simulator of the subprotocol π_{idua}, say $\mathcal{S}_{\pi_{\mathsf{idua}}}$ (running an ideal-world evaluation of functionality $\mathcal{F}_{\mathsf{idua}}$, refer Fig 14).

To complete the proof, we argue how each of the simulators maintain that the view of $\mathcal{A}$ in the ideal world is indistinguishable from its view in the real-world.

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$ corresponding to $t_a = 0$. The only difference between the real and the ideal world is that the messages of Round 1 and 2 of first instance of π_{idua} and Round 1 of the other instances of π_{idua} are obtained via the $\mathcal{S}_{\pi_{\mathsf{idua}}}$. To argue indistinguishability, we define a sequence of intermediate hybrids.

$\textsc{hyb}_0$: Same as $\textsc{real}_{\pi_{\mathsf{god}}^{\mathsf{dyn}},\mathcal{A}}$.

$\textsc{hyb}_1$: Same as $\textsc{hyb}_0$ except that the messages in Round 1 and 2 of the first instance of π_{idua} are simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$.

$\textsc{hyb}_\ell$ ($\ell = 2$ to $\lceil n/2 \rceil$) is defined as -

$\textsc{hyb}_\ell$: Same as $\textsc{hyb}_{\ell-1}$ except that the Round 1 message of the ℓth instance of π_{idua} is simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$.

Note that $\textsc{hyb}_{\lceil n/2 \rceil} = \textsc{ideal}_{\mathcal{F}_{\mathsf{god}},\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}}$. Indistinguishability between each consecutive pair of hybrids follows from security of π_{idua}, proving indistinguishability of $\mathcal{A}$'s view in the real-world and ideal-world.

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ corresponding to $t_a \geq 1$. We argue that the view of $\mathcal{A}$ in the ideal world is indistinguishable from its view in the real-world via a series of intermediate hybrids.

$\textsc{hyb}_0$: Same as $\textsc{real}_{\pi_{\mathsf{god}}^{\mathsf{dyn}},\mathcal{A}}$.

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$

Let H and Corr denote the set of *indices* of honest parties (in $\mathcal{P} \setminus \mathcal{E}$) and parties in $\mathcal{E}$ respectively.

Let $\mathcal{S}_{\pi_{\mathsf{idua}}}$ denote the simulator of the function-delayed protocol π_{idua} realizing $\mathcal{F}_{\mathsf{idua}}$ which comprises of the following algorithms: **(a)** $\mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i)$ to compute the Round 1 simulated message sent on behalf of an honest P_i using randomness r_i and **(b)** $\mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}'_1, y', f', \mathcal{L}'; r_i)$ to compute the Round 2 simulated message sent on behalf of an honest P_i where T'_1 denotes the transcript of Round 1, y' denotes the output obtained from the ideal functionality $\mathcal{F}_{\mathsf{idua}}$ computing f' and $\mathcal{L}'$ denotes the set of participants. The following steps are carried out by $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$:

– **Step 1:** For $k = 1$ to $\lceil n/2 \rceil$, let $\mathsf{msg}_i^{1,k} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i^k)$ ($i \in \mathsf{H}$) correspond to the Round 1 message of P_i for the kth instance obtained by invoking $\mathcal{S}_{\pi_{\mathsf{idua}}}$ with fresh independent randomness r_i^k. Note that this message is independent of the function to be computed by π_{idua} and the number of parties. Send $\{\mathsf{msg}_i^{1,k}\}_{k \in [\lceil n/2 \rceil]}$ on behalf of P_i to $\mathcal{A}$. Receive $\{\mathsf{msg}_j^{1,k}\}_{k \in [\lceil n/2 \rceil]}$ sent by P_j ($P_j \in \mathcal{E}$).

– **Step 2:** Let $\mathsf{T}_1^1 = \{\mathsf{msg}_j^{1,1}\}_{P_j \in \mathcal{E}}$, using which $\mathcal{S}_{\pi_{\mathsf{idua}}}$ returns the extracted input $\{x_j^1\}_{P_j \in \mathcal{E}}$ and invokes the ideal functionality $\mathcal{F}_{\mathsf{idua}}$ on behalf of $\mathcal{A}$.

Step 2.1: If $n = 1, 2$, invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j^1\}_{P_j \in \mathcal{E}}$ on behalf of the adversary. Receive an output value y in return, which is forwarded to $\mathcal{S}_{\pi_{\mathsf{idua}}}$ as response from $\mathcal{F}_{\mathsf{idua}}$. Send $\mathsf{msg}_i^{2,1} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^1, y, f, \mathcal{P}; r_i^1)$ ($i \in \mathsf{H}$) on behalf of P_i to $\mathcal{A}$. This completes the simulation for this case.

Step 2.2: Else, the following steps are run:

- Invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j^1\}_{P_j \in \mathcal{E}}$ on behalf of the adversary. Receive an output value y in return. Compute $(L_1, \ldots L_n) = f_{\mathsf{LSh}}^{\alpha', \beta'}(y)$ (Fig 1) where $\alpha' = n, \beta' = \lfloor n/2 \rfloor$. Return $y' = \{L_j\}_{P_j \in \mathcal{E}}$ as response of $\mathcal{F}_{\mathsf{idua}}$ to $\mathcal{S}_{\pi_{\mathsf{idua}}}$. Send $\mathsf{msg}_i^{2,1} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^1, y, f, \mathcal{P}; r_i^1)$ ($i \in \mathsf{H}$) on behalf of P_i to $\mathcal{A}$.
- Recall that in Round r of $\mathsf{LRec}^{n-2, \lfloor \frac{n}{2} \rfloor}$ ($r \in [1, \lceil n/2 \rceil - 1]$), summand s_{n-r-1} is attempted to be reconstructed. $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$ does the following in Round r ($r \in [1, \lceil n/2 \rceil - 1]$) of $\mathsf{LRec}^{n-2, \lfloor \frac{n}{2} \rfloor}$: Send $(s_{il}, o_{il}) \in L_l$ on behalf of P_l, $l \in \mathsf{H}$, where $i = n - r - 1$.

Fig. 19: Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},0}$

HYB_1**:** Same as HYB_0 except that the following is done w.r.t messages sent in instance $k = 1$. Messages in Round 1 and 2 of π_{idua} are simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$ and the steps of $\mathsf{LRec}^{\mathsf{n}-2, \lfloor \frac{\mathsf{n}}{2} \rfloor}()$ is simulated identical to corresponding steps in $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$.

More generally, HYB_ℓ for $\ell = 1$ to $\lceil n/2 \rceil$ is defined as:

HYB_ℓ**:** Same as $\mathrm{HYB}_{\ell-1}$ except that the following is done w.r.t messages sent in instance $k = \ell$. Messages in Round 1 and 2 of π_{idua} are simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$ and the steps of $\mathsf{LRec}^{\mathsf{n}-2, \lfloor \frac{\mathsf{n}}{2} \rfloor}()$ is simulated identical to corresponding steps in $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$.

Since $\mathrm{HYB}_{\lceil n/2 \rceil} = \mathrm{IDEAL}_{\mathcal{F}_{\mathsf{god}}, \mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}}$, we show that every two consecutive hybrids are computationally indistinguishable to complete the proof. We claim that $\mathrm{HYB}_{\ell-1} \approx \mathrm{HYB}_\ell$ holds for $\ell = 1$ to $\lceil n/2 \rceil$. Following are the differences between

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$

Let H and Corr denote the set of *indices* of honest parties (in $\mathcal{P} \setminus \mathcal{E}$) and parties in $\mathcal{E}$ respectively.
Let $\mathcal{S}_{\pi_{\mathsf{idua}}}$ denote the simulator of the function-delayed protocol π_{idua} realizing $\mathcal{F}_{\mathsf{idua}}$ which comprises of algorithms $\mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i)$ and $\mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}'_1, y', f', \mathcal{L}'; r_i)$ as described in Fig 19. The following steps are carried out by $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ (where $t_a \geq 1$).

- **Step 1:** For $k = 1$ to $\lceil n/2 \rceil$, let $\mathsf{msg}_i^{1,k} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i^k)$ ($i \in \mathsf{H}$) correspond to the Round 1 message of P_i for the kth instance obtained by invoking $\mathcal{S}_{\pi_{\mathsf{idua}}}$ with fresh independent randomness r_i^k. Note that this message is independent of the function to be computed by π_{idua} and the number of parties. Send $\{\mathsf{msg}_i^{1,k}\}_{k \in [\lceil n/2 \rceil]}$ on behalf of P_i to $\mathcal{A}$. Receive $\{\mathsf{msg}_j^{1,k}\}_{k \in [\lceil n/2 \rceil]}$ sent by P_j ($P_j \in \mathcal{E}$).
- **Step 2:** Initialize $k = 1$, $\mathcal{L} = \mathcal{P}$, $\mathcal{C} = \emptyset$, $\mathfrak{n} = n, t'_a = t_a, t'_p = t_p$. Let $f^{\mathcal{C}}$ denote the function same as f except with default inputs hardcoded for parties in $\mathcal{C}$. Let $\mathsf{T}_1^k = \{\mathsf{msg}_j^{1,k}\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$, using which $\mathcal{S}_{\pi_{\mathsf{idua}}}$ returns the extracted input $\{x_j^k\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ and invokes the ideal functionality $\mathcal{F}_{\mathsf{idua}}$ on behalf of $\mathcal{A}$. If $x_j = \perp$, return $(\perp, \mathcal{B} = P_j)$ as output of $\mathcal{F}_{\mathsf{idua}}$.

Step 2.1: If $\mathfrak{n} = 1, 2$, invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j^k\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ on behalf of corrupt parties that are alive and default inputs on behalf of identified actively corrupt parties in $\mathcal{C}$. Receive an output value y in return, which is forwarded to $\mathcal{S}_{\pi_{\mathsf{idua}}}$ as response from $\mathcal{F}_{\mathsf{idua}}$. Send $\mathsf{msg}_i^{2,k} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^k, y, f^{\mathcal{C}}, \mathcal{L}; r_i^k)$ ($i \in \mathsf{H}$) on behalf of P_i to $\mathcal{A}$. This completes the simulation for this case.

Step 2.2: Else, we have two cases. Let $\alpha' = \mathfrak{n} - 2$ and $\beta' = \lfloor \mathfrak{n}/2 \rfloor$.

- If $t'_a = 0$, invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j^k\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ on behalf of corrupt parties that are alive and default inputs on behalf of identified actively corrupt parties in $\mathcal{C}$. Receive an output value y in return. Compute $(L_1, \ldots L_q) = f_{\mathsf{LSh}}^{\alpha',\beta'}(y)$ (Fig 1) among parties in $\mathcal{L}$ (where $q = |\mathcal{L}|$) and return $y' = \{L_j\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ as response of $\mathcal{F}_{\mathsf{idua}}$ to $\mathcal{S}_{\pi_{\mathsf{idua}}}$.
- Else, for $P_j \in \mathcal{E}$, set $L_j = \left(\{s_{ij}, o_{ij}\}_{i \in [\alpha',\beta']}, \{c_{il}\}_{i \in [\alpha',\beta'], P_l \in \mathcal{L}}\right)$ where s_{ij} ($i \in [\alpha', \beta']$) are randomly chosen and $(c_{ij}, o_{ij}) \leftarrow \mathsf{eCom}(s_{ij}; r_{ij})$ (with trapdoor t) computed as per protocol specifications. $\{c_{il}\}_{i \in [\alpha',\beta'], l \in \mathsf{H}}$ are computed as commitments on dummy values, say involving $\{s'_{il}, o'_{il}\}_{i \in [\alpha',\beta'], l \in \mathsf{H}}$. Return $y' = \{L_j\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ as response of $\mathcal{F}_{\mathsf{idua}}$ to $\mathcal{S}_{\pi_{\mathsf{idua}}}$.
- Send $\mathsf{msg}_i^{2,k} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^k, y', f_{\mathsf{LSh}}^{\alpha',\beta'} \diamond f^{\mathcal{C}}, \mathcal{L}; r_i^k)$ ($i \in \mathsf{H}$) on behalf of P_i to $\mathcal{A}$. Let $\mathsf{T}_2^k = \{\mathsf{msg}_j^{2,k}\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ denote the messages received in Round 2 from P_j.

Fig. 20: Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$

these hybrids: **(a)** While in $\textsc{hyb}_{\ell-1}$, Round 1 - 2 messages of the ℓ^{th} instance of π_{idua} are generated using honest parties' inputs; they are generated via $\mathcal{S}_{\pi_{\mathsf{idua}}}$ in $\textsc{hyb}_{\ell}$. **(b)** The steps of $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor \frac{\mathfrak{n}}{2} \rfloor}()$ in instance ℓ are simulated as per $\mathcal{S}_{\mathsf{fair}}^{\mathsf{dyn},t_a}$ in $\textsc{hyb}_{\ell}$. Indistinguishability follows from security of π_{idua} and the security argument in Appendix B.1, which holds except with negligible probability. Since there are only polynomially-many hybrids (i.e. $\lceil n/2 \rceil$) between $\textsc{real}_{\pi_{\mathsf{god}}^{\mathsf{dyn}},\mathcal{A}}$ and

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ *(Contd)*

There are 2 cases based on whether $\mathcal{A}$ aborts the computation of π_{idua}.

- If either **(a)** $\mathcal{S}_{\pi_{\mathsf{idua}}}$ invokes $\mathcal{F}_{\mathsf{idua}}$ with $(\mathtt{abort}, \mathcal{B})$ with $|\mathcal{B}| \geq 1$ upon using $(\mathsf{T}_1^k, \mathsf{T}_2^k)$ or **(b)** $(\perp, \mathcal{B})$ had been returned as output of $\mathcal{F}_{\mathsf{idua}}$ to $\mathcal{A}$, then $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ does the following - Update $\mathcal{C} = \mathcal{C} \cup \mathcal{B}$, $\mathcal{L} = \mathcal{L} \setminus \mathcal{B}$, $\mathfrak{n} = \mathfrak{n} - 2|\mathcal{B}|$, $k = k + 1$, $t'_a = t'_a - |\mathcal{B}|$, $t'_p = t'_p - |\mathcal{B}|$ and repeat this simulation of step 2 using updated value of $\mathfrak{n}, k, t'_a, t'_p$ and the updated sets.
- Else, if $\mathcal{S}_{\pi_{\mathsf{idua}}}$ invokes $\mathcal{F}_{\mathsf{idua}}$ with $\mathtt{continue}$ upon using $(\mathsf{T}_1^k, \mathsf{T}_2^k)$, run the following steps to simulate $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$. Recall that in Round r of $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$ ($r \in [1, \lceil \mathfrak{n}/2 \rceil - 1]$), summand $s_{\mathfrak{n}-r-1}$ is attempted to be reconstructed. If $t'_a = 0$, send $(s_{il}, o_{il}) \in L_l$ on behalf of P_l, $l \in \mathsf{H}$, where $i = \mathfrak{n} - r - 1$ in Round r ($r \in [1, \lceil \mathfrak{n}/2 \rceil - 1]$); completing the simulation. Else, (when $t'_a \neq 0$) $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ does the following in Round r of $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$.
  1. If $r \leq \mathfrak{n} - t'_p - 2$: Let $i = \mathfrak{n} - r - 1$. Send $\{s'_{il}, o'_{il}\}_{l \in \mathsf{H}}$ on behalf of honest parties and receive (s'_{ij}, o'_{ij}) from each $P_j \in \mathcal{L} \cap \mathcal{E}$. Initialize $\mathsf{Val}_i = \mathcal{L} \setminus \mathcal{E}$. Add $P_j \in \mathsf{Val}_i$ if $P_j \in \mathcal{E}$ sends $(s'_{ij}, o'_{ij}) = (s_{ij}, o_{ij})$ (consistent with L_j returned as output of $\mathcal{F}_{\mathsf{idua}}$ to P_j). If $|\mathsf{Val}_i| < i + 1$, then let $\mathcal{B} = \mathcal{L} \setminus \mathsf{Val}_i$. Update $\mathcal{C} = \mathcal{C} \cup \mathcal{B}$, $\mathcal{L} = \mathcal{L} \setminus \mathcal{B}$, $\mathfrak{n} = \mathfrak{n} - 2|\mathcal{B}|$, $k = k + 1$, $t'_a = t'_a - |\mathcal{B}|$, $t'_p = t'_p - |\mathcal{B}|$ and repeat the simulation of step 2 using these updated values.
  2. If $r = \mathfrak{n} - t'_p - 1$: This round involves reconstruction of summand $s_{t'_p}$. $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ does the following:
     - ○ Invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j^k\}_{P_j \in \mathcal{L} \cap \mathcal{E}}$ on behalf of corrupt parties that are alive and default inputs on behalf of identified actively corrupt parties in $\mathcal{C}$. Receive output value y in return.
     - ○ Note that reconstruction of summands $s_{t'_p+1} \ldots s_{\mathfrak{n}-2}$ has been completed and the summands s_i, where $i \in [\lfloor \mathfrak{n}/2 \rfloor, \ldots t'_p - 1]$ is already fully determined by values returned as output of $\mathcal{F}_{\mathsf{idua}}$. Compute $s_{t'_p} = y - \sum_{i=\lfloor \mathfrak{n}/2 \rfloor}^{t'_p-1} s_i - \sum_{i=t'_p+1}^{\mathfrak{n}-2} s_i$.
     - ○ Let $\mu = t'_p$. Interpolate a μ-degree polynomial $g_\mu(x)$ satisfying $g_\mu(0) = s_\mu$ and $g_\mu(j) = s_{\mu j}$ for $P_j \in \mathcal{E} \cap \mathcal{L}$. Let $s_{\mu l} = g_\mu(l)$ for $l \in \mathsf{H}$. Compute $o_{\mu l} \leftarrow \mathsf{Equiv}(c_{\mu l}, (s'_{\mu l}, o'_{\mu l}), s_{\mu l}, t)$. Send $(s_{\mu l}, o_{\mu l})$ on behalf of P_l, $l \in \mathsf{H}$.
  3. If $r \in [(\mathfrak{n} - t'_p), \lceil \mathfrak{n}/2 \rceil - 1]$: Send (s'_{il}, o'_{il}) on behalf of P_l, $l \in \mathsf{H}$, where $i = \mathfrak{n} - r - 1$.

Fig. 21: Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}$ *(Contd)*

$\textsc{ideal}_{\mathcal{F}_{\mathsf{god}}, \mathcal{S}_{\mathsf{god}}^{\mathsf{dyn},t_a}}$, we conclude that the view of $\mathcal{A}$ in the real-world is computationally indistinguishable from its view in the ideal world.

□

C Proofs of Upper Bounds for Boundary Corruption

C.1 Proof of Security of $\pi^{\mathsf{bou}}_{\mathsf{god}}$ (Theorem 6)

Proof. We prove Theorem 6 by presenting two separate simulators $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$ and $\mathcal{S}^{\mathsf{mal}}_{\mathsf{god}}$ for the case of pure passive corruption $(t_a, t_p) = (0, n-1)$ and $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ involving active corruptions respectively. The protocol $\pi^{\mathsf{bou}}_{\mathsf{god}}$ is analyzed in a $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$- hybrid model where the parties have access to a trusted party computing $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$ (Fig. 8). Additionally, the simulator of the subprotocol π_{god}, say $\mathcal{S}_{\pi_{\mathsf{god}}}$ is also invoked.

Simulator $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$ wrt $(t_a, t_p) = (0, n-1)$: Let $\mathcal{A}$ be a boundary-admissible adversary with parameters $(t_a, t_p) = (0, n-1)$ in the $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$-hybrid model execution of $\pi^{\mathsf{bou}}_{\mathsf{god}}$ (hybrid-world). The simulator $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{god}}$ (refer Fig 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 22. It is straightforward to see that the view of $\mathcal{A}$ in the ideal world is indistinguishable from the view of $\mathcal{A}$ in the $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$-hybrid model execution of $\pi^{\mathsf{bou}}_{\mathsf{god}}$. The only difference is that in the ideal world, Round 1 of $\pi^{\mathsf{bou}}_{\mathsf{god}}$ is obtained via $\mathcal{S}_{\pi_{\mathsf{god}}}$, whose simulation is independent of the parties' inputs. We can thus conclude that $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$ outputs a view indistinguishable to the view of $\mathcal{A}$ in the hybrid-world.

Simulator $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$

Let $\mathsf{Corr} \subset [n]$ and H be the set of indices of corrupt and honest parties respectively. Suppose $\mathcal{S}_{\pi_{\mathsf{god}}}$ denote the simulator of the subprotocol π_{god}. Let $\mathcal{S}^{1,i}_{\pi_{\mathsf{god}}}(r_i)$ denote the algorithm to compute the Round 1 simulated message of honest P_i using randomness r_i. The following steps are carried out by $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$:

- *Simulation of Round 1 of π_{god}:* Send $\mathsf{msg}^1_i \leftarrow \mathcal{S}^{1,i}_{\pi_{\mathsf{god}}}(r_i)$ $(i \in \mathsf{H})$ on behalf of P_i. Recall that this step is independent of parties' inputs. Receive msg^1_j as Round 1 message of π_{god} sent by P_j $(j \in \mathsf{Corr})$.
- *Invoking $\mathcal{F}_{\mathsf{god}}$:* Receive $\{x_i\}_{i \in \mathsf{Corr}}$ corresponding to the parties controlled by passive adversary $\mathcal{A}$. Invoke $\mathcal{F}_{\mathsf{god}}$ on behalf of $\mathcal{A}$ with $\{x_i\}_{i \in \mathsf{Corr}}$ to receive an output value y in return.
- *Output of $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$ to $\mathcal{A}$:* Compute the authenticated secret-sharing of value y with threshold $\lfloor n/2 \rfloor$ as $(S_1 \ldots S_n) = f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}}(y)$ (Figure 6) and send $S_j = \left(\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{i \in [n]}\right)$ as output of $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$ to P_j $(j \in \mathsf{Corr})$.
- *Round 3:* For each $i \in \mathsf{H}$, send $(\mathsf{sh}_i, \mathsf{tag}_{ij})$ $(j \neq i)$ on behalf of P_i.

Fig. 22: Simulator $\mathcal{S}^{\mathsf{sh}}_{\mathsf{god}}$

Simulator $\mathcal{S}^{\mathsf{mal}}_{\mathsf{god}}$ wrt $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$: Let $\mathcal{A}$ be a boundary-admissible malicious adversary with corruption parameters $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ in

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$

Let $\mathsf{Corr} \subset [n]$ and $\mathsf{H} = [n] \setminus \mathsf{Corr}$ be the set of indices of the parties controlled by adversary and the honest parties respectively.
Let $\mathcal{S}_{\pi_{\mathsf{god}}}$ denote the simulator of the 3-round subprotocol π_{god} whose Round 1 is function and input independent. Let $\mathcal{S}_{\pi_{\mathsf{god}}}^{1,i}(r_i), \mathcal{S}_{\pi_{\mathsf{god}}}^{2,i}(\mathsf{T}_1, f'; r_i)$ and $\mathcal{S}_{\pi_{\mathsf{god}}}^{3,i}(\mathsf{T}_2, y'; r_i)$ denote the algorithms to compute simulated message of honest P_i (using randomness r_i) corresponding to Round 1, 2 and 3 respectively. Here, $\mathsf{T}_1, \mathsf{T}_2$, f', y' refer to protocol transcript of Round 1, protocol transcript until Round 2, function to be computed and the output from its ideal functionality respectively. The following steps are carried out by $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$:

- *Simulation of Round 1 of* π_{god}*:* Send $\mathsf{msg}_i^1 \leftarrow \mathcal{S}_{\pi_{\mathsf{god}}}^{1,i}(r_i)$ $(i \in \mathsf{H})$ on behalf of P_i. Recall that this step is independent of parties' inputs. Receive msg_j^1 as Round 1 message of π_{god} sent by P_j $(j \in \mathsf{Corr})$ and set $\mathsf{T}_1 = \{\mathsf{msg}_j^1\}_{j \in \mathsf{Corr}}$.
- *Interaction of* $\mathcal{A}$ *with* $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$*:* Receive $\{x_i\}_{i \in \mathsf{Corr}}$ sent by $\mathcal{A}$ to $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$. If for any $i \in \mathsf{Corr}$, x_i is outside of domain of input, return $\mathcal{B} = P_i$ (identified cheater) as output of $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ to $\mathcal{A}$ and skip to simulation step of *Handling Abort*. Else run the following steps.
  - *Output of* $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ *to* $\mathcal{A}$*:* Choose random sh_j for $j \in \mathsf{Corr}$ and compute its authentication (Step 2, 3 of $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}$ in Fig 6). The resulting values $S_j = \{\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{i \in [n]}\}$ are given to $\mathcal{A}$ as the outputs of the corrupted parties from functionality $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$. Note that functionality $\mathcal{F}_{\mathsf{god}}$ computing f has not been invoked yet.
  - If $\mathcal{A}$ invokes $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ with $(\mathtt{abort}, \mathcal{B})$, proceed to simulation step of *Handling Abort*.
  - *Round 3 in case of no abort:* Else, if $\mathcal{A}$ invokes $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$ with $\mathtt{continue}$, then invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ on behalf of $\mathcal{A}$ to obtain output y. The following steps are used to simulate Round 3:
    1. Interpolate a $\lfloor n/2 \rfloor$-degree polynomial $A(x)$ with $A(j) = \mathsf{sh}_j$ for $j \in \mathsf{Corr}$ and $A(0) = y$.
    2. Set $\mathsf{sh}_i = A(i)$ for $i \in \mathsf{H}$. Using k_{ij} (consistent with output of $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$), compute $\mathsf{tag}_{ij} = \mathsf{Mac}_{k_{ij}}(\mathsf{sh}_i)$. Send $(\mathsf{sh}_i, \mathsf{tag}_{ij})$ $(j \neq i)$ on behalf of P_i in Round 3.
- *Handling Abort.* $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$ does the following:
  - *Round 3:* Send $\mathsf{msg}_i^2 \leftarrow \mathcal{S}_{\pi_{\mathsf{god}}}^{2,i}(\mathsf{T}_1, f^{\mathcal{B}}; r_i)$ $(i \in \mathsf{H})$ on behalf of P_i. Receive msg_j^2 as Round 2 message of π_{god} sent by P_j $(j \in \mathsf{Corr})$. When $\mathcal{S}_{\pi_{\mathsf{god}}}$ returns the extracted input $\{x'_j\}_{j \in \mathsf{Corr}}$ of the corrupt party to invoke its ideal functionality $\mathcal{F}_{\mathsf{god}}$, $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$ invokes $\mathcal{F}_{\mathsf{god}}$ with input $\{x'_j\}_{j \in \mathsf{Corr} \setminus \mathcal{B}}$ on behalf of corrupt P_j (not identified among set of cheaters) and default input on behalf of parties in $\mathcal{B}$. Then, forward the obtained output y' as response to $\mathcal{S}_{\pi_{\mathsf{god}}}$.
  - *Round 4:* Send $\mathsf{msg}_i^3 \leftarrow \mathcal{S}_{\pi_{\mathsf{god}}}^{3,i}(\mathsf{T}_2, y'; r_i)$ $(i \in \mathsf{H})$ on behalf of P_i where $\mathsf{T}_2 = \{\mathsf{msg}_j^1, \mathsf{msg}_j^2\}_{j \in \mathsf{Corr}}$.

Fig. 23: Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$

the $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{god}}^{\mathsf{bou}}$. The simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{god}}$ (refer Figure 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 23.

There are 2 different scenarios based on whether $\mathcal{A}$ aborts the computation of $\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$. In case abort doesn't occur, it follows directly from the properties of privacy of authenticated sharing (Lemma 14) that the view in the ideal world is indistinguishable to the view of $\mathcal{A}$ in the hybrid-world ($\mathcal{F}_{\mathsf{idua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{god}}^{\mathsf{bou}}$). Regarding Round 1 of π_{god} which is computed using honest inputs in the hybrid world but obtained via $\mathcal{S}_{\pi_{\mathsf{god}}}$ in the ideal world, indistinguishability follows from security of π_{god}. Lastly, in this abort case, we note that the only difference between the ideal and hybrid-execution is that the messages of π_{god} are obtained via the simulator $\mathcal{S}_{\pi_{\mathsf{god}}}$ in the former. Indistinguishability thus follows from the security of subprotocol π_{god}. This completes the proof.

□

C.2 Proof of Security of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ (Theorem 7)

Proof. We prove Theorem 7 by presenting two separate simulators $\mathcal{S}_{\mathsf{god}}^{\mathsf{sh},1}$ and $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$ for the case of $(t_a, t_p) = (0, n-1)$ and $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$ respectively. $\mathcal{S}_{\mathsf{god}}^{\mathsf{sh},1}$ and $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$ invoke the simulator of the subprotocol π_{idua}, say $\mathcal{S}_{\pi_{\mathsf{idua}}}$ (running an ideal-world evaluation of functionality $\mathcal{F}_{\mathsf{idua}}$, refer Fig 14).

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{sh},1}$ wrt $(t_a, t_p) = (0, n-1)$: Let $\mathcal{A}$ be a boundary-admissible passive adversary with parameters $(t_a, t_p) = (0, n-1)$ in the execution of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$. The simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{sh},1}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{god}}$ (refer Figure 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 24.

To argue indistinguishability, we note that the only difference between the real and the ideal world is that in the ideal world, the simulator $\mathcal{S}_{\pi_{\mathsf{idua}}}$ is invoked to simulate messages of honest parties in both rounds of the first instance of π_{idua} and Round 1 of the second instance of π_{idua}. We define an intermediate hybrid between the real and ideal world, where only the messages of the first instance of π_{idua} is simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$. It is easy to see that each pair of consecutive hybrids is indistinguishable, following security of π_{idua}. We can thus conclude that the view of $\mathcal{A}$ in the ideal world is indistinguishable to the view of $\mathcal{A}$ in the real world execution of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$.

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$ wrt $(t_a, t_p) = (1, \lfloor n/2 \rfloor)$: Let $\mathcal{A}$ be a malicious adversary controlling at most 1 party actively and upto $\lfloor n/2 \rfloor$ parties passively in an execution of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$. The simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{god}}$ (refer Figure 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 25.

There are 2 different scenarios based on whether $\mathcal{A}$ aborts the computation in first instance of π_{idua}. In case abort doesn't occur, simulation proceeds similar

Simulator $\mathcal{S}^{\mathsf{sh},1}_{\mathsf{god}}$

Let $\mathsf{Corr}, \mathsf{H}$ be the set of indices of corrupt and honest parties respectively. Suppose $\mathcal{S}_{\pi_{\mathsf{idua}}}$ denote the simulator of the subprotocol π_{idua}. Let $\mathcal{S}^{1,i}_{\pi_{\mathsf{idua}}}(r_i)$ and $\mathcal{S}^{2,i}_{\pi_{\mathsf{god}}}(\mathsf{T}_1, y', f', \mathcal{P}'; r_i)$ denote the algorithms to compute simulated message of honest P_i (using randomness r_i) corresponding to Round 1 and 2 respectively. Here, T_1, y', f' and $\mathcal{P}'$ refer to protocol transcript of Round 1, output from its ideal functionality $\mathcal{F}_{\mathsf{idua}}$, function to be computed and the set of participants respectively. The following steps are carried out by $\mathcal{S}^{\mathsf{sh},1}_{\mathsf{god}}$:

- *Invoking $\mathcal{F}_{\mathsf{god}}$:* Receive $\{x_i\}_{i \in \mathsf{Corr}}$ corresponding to the parties controlled by semi-honest adversary $\mathcal{A}$. Invoke $\mathcal{F}_{\mathsf{god}}$ on behalf of $\mathcal{A}$ with $\{x_i\}_{i \in \mathsf{Corr}}$ to receive an output value y in return.
- *Interaction with $\mathcal{S}_{\pi_{\mathsf{idua}}}$:* Compute the authenticated secret-sharing of value y with threshold $\lfloor n/2 \rfloor$ as $(S_1 \ldots S_n) = f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}}(y)$ (Figure 6).
  Round 1: For $k = 1, 2$, let $\mathsf{msg}^{1,k}_i \leftarrow \mathcal{S}^{1,i}_{\pi_{\mathsf{idua}}}(r^k_i)$ $(i \in \mathsf{H})$, where r^1_i, r^2_i are sampled independently. Send $\{\mathsf{msg}^{1,k}_i\}_{k \in [2]}$ on behalf of P_i. Receive $\{\mathsf{msg}^{1,k}_j\}_{k \in [2]}$ sent by P_j $(j \in \mathsf{Corr})$.
  When $\mathcal{S}_{\pi_{\mathsf{idua}}}$ invokes its ideal functionality $\mathcal{F}_{\mathsf{idua}}$ computing $f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}} \diamond f$ with input x_i, send output $S_j = \big(\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{i \in [n]}\big)$ corresponding to P_j $(j \in \mathsf{Corr})$ as response from $\mathcal{F}_{\mathsf{idua}}$.
  Round 2: Send $\mathsf{msg}^{2,1}_i \leftarrow \mathcal{S}^{2,i}_{\pi_{\mathsf{idua}}}(\mathsf{T}_1, y', f^{\lfloor n/2 \rfloor}_{\mathsf{ASh}} \diamond f, \mathcal{P}; r^1_i)$ $(i \in \mathsf{H})$ on behalf of P_i, where $\mathsf{T}_1 = \{\mathsf{msg}^{1,1}_j\}_{j \in \mathsf{Corr}}$ and $y' = \{S_j\}_{j \in \mathsf{Corr}}$ i.e. output of $\mathcal{F}_{\mathsf{idua}}$. Receive $\mathsf{msg}^{2,1}_j$ sent by P_j $(j \in \mathsf{Corr})$.
- *Round 3:* For each $i \in \mathsf{H}$, send $(\mathsf{sh}_i, \mathsf{tag}_{ij})$ $(j \neq i)$ on behalf of P_i.

Fig. 24: Simulator $\mathcal{S}^{\mathsf{sh},1}_{\mathsf{god}}$

to $\mathcal{S}^{\mathsf{mal}}_{\mathsf{god}}$ (Figure 23) except that instead of analysis in $\mathcal{F}^{\mathsf{ASh}}_{\mathsf{idua}}$- hybrid model, the simulator $\mathcal{S}_{\pi_{\mathsf{idua}}}$ is invoked for simulation in Round 1 and Round 2. Another difference is that an additional instance of Round 1 of π_{idua} is simulated. Thus, in case of no abort, it follows from the security argument of $\mathcal{S}^{\mathsf{mal}}_{\mathsf{god}}$ and the security of π_{idua} that the view of $\mathcal{A}$ in the ideal world is indistinguishable to the view of $\mathcal{A}$ in the execution of $\pi^{\mathsf{bou},1}_{\mathsf{god}}$.

Consider case of abort which returns the identity of cheater, say singleton set $\mathcal{B}$. The difference between the ideal and the real execution is that the messages of honest parties are obtained via $\mathcal{S}_{\pi_{\mathsf{idua}}}$ in the ideal world for both instances of π_{idua}. We define an intermediate hybrid between the real and ideal world, where only the messages of the first instance of π_{idua} is simulated using $\mathcal{S}_{\pi_{\mathsf{idua}}}$. It is easy to see that each pair of consecutive hybrids is indistinguishable, following security of π_{idua}. We can thus conclude that the view of $\mathcal{A}$ in the ideal world is indistinguishable to the view of $\mathcal{A}$ in the execution of $\pi^{\mathsf{bou},1}_{\mathsf{god}}$. This completes the proof. □

Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$

Let $\mathsf{Corr} \subset [n]$ and $\mathsf{H} = [n] \setminus \mathsf{Corr}$ be the set of indices of the corrupt parties and the honest parties respectively.
Suppose $\mathcal{S}_{\pi_{\mathsf{idua}}}$ denote the simulator of the subprotocol π_{idua}. Let $\mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i)$ and $\mathcal{S}_{\pi_{\mathsf{god}}}^{2,i}(\mathsf{T}_1, y', f', \mathcal{P}'; r_i)$ denote the algorithms to compute simulated message of honest P_i (using randomness r_i) corresponding to Round 1 and 2 respectively. Here, T_1, y', f' and $\mathcal{P}'$ refer to protocol transcript of Round 1, output from its ideal functionality $\mathcal{F}_{\mathsf{idua}}$, function to be computed and the set of participants respectively. The following steps are carried out by $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$:

– *Round 1:* For $k = 1, 2$, let $\mathsf{msg}_i^{1,k} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{1,i}(r_i^k)$ $(i \in \mathsf{H})$, where r_i^1, r_i^2 are sampled independently. Send $\{\mathsf{msg}_i^{1,k}\}_{k \in 2}$ on behalf of P_i to $\mathcal{A}$. Receive $\{\mathsf{msg}_j^{1,k}\}_{k \in [2]}$ sent by P_j $(j \in \mathsf{Corr})$.

– *Round 2:* Send $\mathsf{msg}_j^{1,1}$ to $\mathcal{S}_{\pi_{\mathsf{idua}}}$ on behalf of P_j for $j \in \mathsf{Corr}$. When $\mathcal{S}_{\pi_{\mathsf{idua}}}$ returns the extracted input $\{x_j\}_{j \in \mathsf{Corr}}$ of the corrupt party to invoke its ideal functionality $\mathcal{F}_{\mathsf{idua}}$ computing $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f$, $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$ does the following:
  - If there exists a $j \in \mathsf{Corr}$ such that $x_j = \perp$, send $y' = (\perp, P_j)$ to $\mathcal{A}$ as output response of $\mathcal{F}_{\mathsf{idua}}$.
  - Else choose random sh_j for $j \in \mathsf{Corr}$ and compute its authentication (Step 2, 3 of $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}()$ of Fig 6). The resulting values $S_j = \{\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{i \in [n]}\}$ for each $j \in \mathsf{Corr}$ are given to $\mathcal{A}$ as the output from functionality $\mathcal{F}_{\mathsf{idua}}$.

  Send $\mathsf{msg}_i^{2,1} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^1, y', f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f, \mathcal{P}; r_i^1)$ $(i \in \mathsf{H})$ on behalf of P_i to $\mathcal{A}$, where $\mathsf{T}_1^1 = \{\mathsf{msg}_j^{1,1}\}_{j \in \mathsf{Corr}}$ and $y' = \{S_j\}_{j \in \mathsf{Corr}}$. Receive $\mathsf{msg}_j^{2,1}$ sent by P_j $(j \in \mathsf{Corr})$ and send it to $\mathcal{S}_{\pi_{\mathsf{idua}}}$ on behalf of P_j.

– *Round 3:* We now have 2 cases -
  - If $\mathcal{S}_{\pi_{\mathsf{idua}}}$ invokes $\mathcal{F}_{\mathsf{idua}}$ with $(\mathtt{abort}, \mathcal{B})$, do the following: Send $\mathsf{msg}_j^{1,2}$ to $\mathcal{S}_{\pi_{\mathsf{idua}}}$ on behalf of $\mathcal{A}$ $(P_j \in \mathcal{E})$ corresponding to second instance of π_{idua}. Suppose $\{x_j\}_{j \in \mathsf{Corr}}$ is the extracted input returned by $\mathcal{S}_{\pi_{\mathsf{idua}}}$, then invoke $\mathcal{F}_{\mathsf{god}}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ and substituting default input of party in $\mathcal{B}$; on behalf of $\mathcal{A}$ to obtain output y. Send $\mathsf{msg}_i^{2,2} \leftarrow \mathcal{S}_{\pi_{\mathsf{idua}}}^{2,i}(\mathsf{T}_1^2, y, f^{\mathcal{B}}, \mathcal{P} \setminus \mathcal{B}; r_i^2)$ $(i \in \mathsf{H})$ on behalf of P_i to $\mathcal{A}$, where $\mathsf{T}_1^2 = \{\mathsf{msg}_j^{1,2}\}_{j \in \mathsf{Corr}}$.
  - If $\mathcal{S}_{\pi_{\mathsf{idua}}}$ invokes $\mathcal{F}_{\mathsf{idua}}$ with $\mathtt{continue}$, run the same steps as Round 3 simulation in case of no abort of $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$ (Fig 23).

Fig. 25: Simulator $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal},1}$

C.3 Proof of Security of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$ (Theorem 8)

We prove Theorem 8 by presenting two separate simulators $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$ and $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$ for the case of corruption scenarios $(t_a, t_p) = (0, n-1)$ and $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ respectively. The protocol $\pi_{\mathsf{fair}}^{\mathsf{bou}}$ is analyzed in a $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model where the parties have access to a trusted party computing $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ (Fig 11).

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$ wrt $(t_a, t_p) = (0, n-1)$: Let $\mathcal{A}$ be the boundary-admissible passive adversary controlling upto $(n-1)$ parties in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$. The simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$, running an ideal-world evaluation of the functionality

$\mathcal{F}_{\mathsf{god}}$ (refer Fig 16) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 26. It is straightforward to see that the view of $\mathcal{A}$ in the ideal world is identical to the view of $\mathcal{A}$ in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$.

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$

Let $\mathsf{Corr} \subset [n]$, H denote the set of indices of corrupt and honest parties respectively. The following steps are carried out by $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$:

- *Invoking $\mathcal{F}_{\mathsf{god}}$:* Receive $\{x_j\}_{j \in \mathsf{Corr}}$ corresponding to the parties controlled by passive adversary $\mathcal{A}$. Invoke $\mathcal{F}_{\mathsf{god}}$ on behalf of $\mathcal{A}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ to receive an output value y in return.
- *Output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ to $\mathcal{A}$:* Compute the authenticated secret-sharing of value y with threshold $t = \lfloor n/2 \rfloor$ (Figure 6) as $(S_1 \ldots S_n) = f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}(y)$ and send $S_j = \left(\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{i \in [n]}\right)$ as output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ to P_j ($j \in \mathsf{Corr}$).
- *Round 3:* Send $(\mathsf{sh}_i, \mathsf{tag}_{ij})$ on behalf of P_i for each $i \in \mathsf{H}, j \neq i$.

Fig. 26: Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{sh}}$

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$

Let $\mathsf{Corr} \subset [n]$ and $\mathsf{H} = [n] \setminus \mathsf{Corr}$ be the set of indices corrupt and honest parties respectively. The following steps are carried out by $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$:

- *Interaction of $\mathcal{A}$ with $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$:* Receive $\{x_j\}_{j \in \mathsf{Corr}}$ sent by malicious $\mathcal{A}$ to $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$. If for any $j \in \mathsf{Corr}$, x_j is outside of domain of input, send $\perp$ as output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ to $\mathcal{A}$ and send $\perp$ as input to $\mathcal{F}_{\mathsf{fair}}$ on behalf of $\mathcal{A}$; completing the simulation. Else run the following steps.
- *Output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ to $\mathcal{A}$:* Choose random sh_j for $j \in \mathsf{Corr}$ and compute its authentication (Step 2, 3 of $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}$ in Fig 6). The resulting values $S_j = \{\mathsf{sh}_j, \{k_{ij}\}_{i \in [n]}, \{\mathsf{tag}_{ji}\}_{j \in [n]}\}$ are given to $\mathcal{A}$ as the outputs of the corrupted parties from functionality $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$.
- *Invoking $\mathcal{F}_{\mathsf{fair}}$:* We have 2 cases based on whether $\mathcal{A}$ invokes $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ with `abort` or `continue`.
  - `abort`: Send $\perp$ as input to $\mathcal{F}_{\mathsf{fair}}$ on behalf of $\mathcal{A}$; thereby completing the simulation.
  - `continue`: Invoke $\mathcal{F}_{\mathsf{fair}}$ with $\{x_j\}_{j \in \mathsf{Corr}}$ on behalf of $\mathcal{A}$ to obtain y.
- *Round 3:* The following steps are used to simulate Round 3 -
  - Interpolate a $\lfloor n/2 \rfloor$-degree polynomial $A(x)$ with $A(j) = \mathsf{sh}_j$ for $j \in \mathsf{Corr}$ and $A(0) = y$.
  - Set $\mathsf{sh}_i = A(i)$ for $i \in \mathsf{H}$. Using k_{ij} (consistent with the output of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$ sent to $\mathcal{A}$), compute $\mathsf{tag}_{ij} = \mathsf{Mac}_{k_{ij}}(\mathsf{sh}_i)$.
  - Send $(\mathsf{sh}_i, \mathsf{tag}_{ij})$ ($i \in \mathsf{H}$) on behalf of P_i ($j \neq i$).

Fig. 27: Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$

Simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$ *wrt* $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$*:* Let $\mathcal{A}$ be a malicious adversary with corruption parameters $(t_a, t_p) = (\lceil n/2 \rceil - 1, \lfloor n/2 \rfloor)$ parties in the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$. The simulator $\mathcal{S}_{\mathsf{fair}}^{\mathsf{mal}}$, running an ideal-world evaluation of the functionality $\mathcal{F}_{\mathsf{fair}}$ (Figure 15) computing f whose behaviour simulates the behaviour of $\mathcal{A}$ is described in Figure 27.

There are 2 different scenarios based on whether $\mathcal{A}$ aborts the computation of $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$. In case of abort, it follows from privacy of sharing function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}$ that the view of $\mathcal{A}$ in the ideal world is indistinguishable to the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$. In case of no abort, the simulation proceeds similar to $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$ (Fig 23, no abort case). We can thus conclude based on the security arguments of $\mathcal{S}_{\mathsf{god}}^{\mathsf{mal}}$ that the view of $\mathcal{A}$ in the ideal world is indistinguishable to the $\mathcal{F}_{\mathsf{ua}}^{\mathsf{ASh}}$-hybrid model execution of $\pi_{\mathsf{fair}}^{\mathsf{bou}}$. This completes the proof.