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 $(\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ to $(0, n-1)$, the boundary corruption restricts the adversary only to the boundary cases of $(\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ 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, $\lceil n/2 \rceil + 1$ 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

[*] Arpita Patra is supported by SERB Women Excellence Award 2017 (DSTO 1706).

1 Introduction

Secure multi-party computation (MPC) [1, 2, 3], 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 atmost minority of the parties (referred to as malicious minority) [4]. 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 [5]. 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 $(0, n-1)$. 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. 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 [6, 7, 8], 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 [9]. 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 [6] shows necessity of 3 rounds for fairness in the plain and CRS setting, 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 [8] extends the necessity of 3 rounds for any t (including $t = 1$) as long as $n/3 < t < n/2$. The work of [7] 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 [10, 11] showed that 2-rounds are sufficient to achieve robustness in the passive-majority setting. In accordance with the impossibility of [4] and sufficiency of honest-majority shown by classical result of [12], the upper bounds in the malicious setting involve $t < n/2$ parties. These include the 3-round constructions of [7, 13, 14] based on tools such as Zaps, multi-key FHE, dense crypto-systems. The protocol of [7] can be collapsed to two rounds given access to a PKI. In the information-theoretic setting involving $t < n/4$ malicious corruptions, the work of [15] 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 [16, 8].

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 [17] 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 [18, 19, 20, 21]. 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 [22] 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* [22, 23, 24, 25, 26, 27, 28], 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 (Infact, boundary-admissible adversary for odd n degenerates to the adversarial model studied in “best-of-both-worlds" MPC [28]). 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 [29, 5]. 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*. 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 corruption 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 [7, 10, 11]. 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 [30, 31, 16] 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 [6, 7, 8]. 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	[9] (private)	[10, 11] (plain)
Malicious-minority	Fair, GOD	3	[7, 8] (public)	[13, 14] (plain)
	Fair, GOD	2	[9] (private)	[7] (private)
Boundary	Fair	**3**	**[This]** (private)	**[This]** (public)
	GOD	**4** (**3** when $t_a \leq 1$)	**[This]** (private)	**[This]** (public)
Dynamic	Fair, GOD	$\mathbf{\lceil \frac{n}{2} \rceil + 1}$	**[This]** (private)	**[This]** (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 [10, 11] 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 failure happens 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 [5] 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 [5] 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 [32]). The GOD upper bound additionally assumes the existence of Non-Interactive Zero-Knowledge (NIZK) in the common random/reference string model.

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 [28], we point that we need to tackle the exact corruption scenarios as that of the protocols of [28] 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. Assumption wise, while both the robust constructions

(a) and **(b)** rely on NIZKs, the former additionally 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 connected by a broadcast channel and a setup where parties have access to 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 Turing (PPT) 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 $\mathcal{A}$ 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 active and passively corrupt parties by $\mathcal{D}$ and $\mathcal{E}$ respectively, where $|\mathcal{D}| = t_a$ and $|\mathcal{E}| = t_p$.

Roadmap. Our lower and upper bounds for dynamic and boundary corruption appear in Sections 3-4 and in Sections 5-6 respectively. The security definitions and proofs appear in the full version [33].

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 in the setting of dynamic corruption.

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 $\mathcal{A}$ with corruption threshold (t_a, t_p) and in the presence of a setup with CRS and PKI. At a high-level, our proof argument defines a sequence of hybrid executions of π, navigating 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. We now prove the sequence of lemmas to complete the proof.

Lemma 1. *In an execution of π where all parties behave honestly upto (and including) Round $(\lceil \frac{n}{2} \rceil - i)$ for $i \in [\lceil \frac{n}{2} \rceil - 1]$, there exists a set of parties S^i with size $(\lfloor \frac{n}{2} \rfloor + i)$ whose combined view at the end of Round $\lceil \frac{n}{2} \rceil - i$ suffices to compute the output.*

Proof. We prove the lemma by induction. Let $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ denote the set of parties and $\mathcal{D}(\mathcal{E})$ denote the set of actively (passively) corrupt parties where $\mathcal{D} \subseteq \mathcal{E}$. Here $|\mathcal{D}| = t_a$ and $|\mathcal{E}| = t_p$.

Base Case ($i = 1$): We consider an execution of the protocol π with a dynamic-admissible adversary $\mathcal{A}$ corrupting parties with threshold $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - 1, \lfloor n/2 \rfloor)$ and an adversarial strategy $\mathcal{A}_1$ as follows. The set of actively corrupt parties $\mathcal{D}$ behave honestly upto (and including) Round $\lceil \frac{n}{2} \rceil - 1$ and simply remain silent in the last round i.e the $\lceil \frac{n}{2} \rceil$th round. Since $\mathcal{A}$ receives all the desired communication throughout the protocol, it follows directly from the correctness of π that $\mathcal{A}$ must be able to compute the output. Since π is assumed to be fair, the honest parties must also be able to compute the output even without the $\lceil \frac{n}{2} \rceil$th round communication from parties in $\mathcal{D}$. We can now conclude that the combined view of parties in $\mathcal{P} \setminus \mathcal{D}$ at the end of Round $\lceil \frac{n}{2} \rceil - 1$ must suffice to compute the output. Thus, the set $S^1 = \mathcal{P} \setminus \mathcal{D}$ of parties with size $n - t_a = n - (\lceil \frac{n}{2} \rceil - 1) = \lfloor \frac{n}{2} \rfloor + 1$ hold a combined view at the end of Round $\lceil \frac{n}{2} \rceil - 1$ that suffices to compute the output. This completes the base case.

Induction Hypothesis ($i = \ell$). Suppose the statement is true for $i = \ell$ i.e. if all parties behave honestly upto (and including) Round $(\lceil \frac{n}{2} \rceil - \ell)$, then there exists a set of parties, say S^ℓ, with $|S^\ell| = (\lfloor \frac{n}{2} \rfloor + \ell)$ whose combined view at the end of $(\lceil \frac{n}{2} \rceil - \ell)$th round, suffices to compute the output.

Induction Step ($i = \ell + 1$). We consider an execution of the protocol π with a dynamic-admissible adversary $\mathcal{A}$ corrupting parties with threshold $(t_a, t_p) = (\lceil \frac{n}{2} \rceil - \ell - 1, \lfloor \frac{n}{2} \rfloor + \ell)$ and $\mathcal{E} = S^\ell$ as defined in the induction hypothesis and an adversarial strategy $\mathcal{A}_{\ell+1}$ as follows. The set of actively corrupt parties $\mathcal{D}$ behave honestly upto (and including) Round $(\lceil \frac{n}{2} \rceil - \ell - 1)$ and simply remain silent from Round $(\lceil \frac{n}{2} \rceil - \ell)$ onwards. Since $\mathcal{A}$ receives all the desired communication upto (and including) Round $(\lceil \frac{n}{2} \rceil - \ell)$ of π (as per an honest execution) on behalf of parties in $\mathcal{E}$, it follows directly from the induction hypothesis that the combined view of the parties in $\mathcal{E}$ where $|\mathcal{E}| = \lfloor \frac{n}{2} \rfloor + \ell$ must suffice to compute the output. Since π is assumed to be fair, the honest parties must also be able to compute the output even though the parties in $\mathcal{D}$ stop communicating from Round $(\lceil \frac{n}{2} \rceil - \ell)$ onwards. We can now conclude that the combined view of parties in $\mathcal{P} \setminus \mathcal{D}$ at the end of Round $(\lceil \frac{n}{2} \rceil - \ell - 1)$ must suffice to compute the output. Thus, the set $S^{\ell+1} = \mathcal{P} \setminus \mathcal{D}$ of parties with size $n - t_a = n - (\lceil \frac{n}{2} \rceil - \ell - 1) = \lfloor \frac{n}{2} \rfloor + \ell + 1$ hold a combined view at the end of Round $(\lceil \frac{n}{2} \rceil - \ell - 1)$ that suffices to compute the output. This completes the induction hypothesis and the proof of Lemma 1. □

Lemma 2. *There exists an adversary $\mathcal{A}$ that is able to compute the output at the end of Round 1 of π.*

Proof. When $i = \lceil \frac{n}{2} \rceil - 1$, Lemma 1 implies that if all parties behave honestly in Round 1, then there exists a set $S^{\lceil \frac{n}{2} \rceil - 1}$ of $(\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil - 1) = n - 1$ parties whose combined view suffices to compute the output at the end of Round 1.

Consequently, a dynamic-admissible adversary $\mathcal{A}$ corrupting the parties with threshold $(t_a, t_p) = (0, n-1)$ and $(\mathcal{D} = \emptyset, \mathcal{E} = S^{\lceil \frac{n}{2} \rceil - 1})$ must be able to compute the output at the end of Round 1 itself. □

Lemma 3. *Protocol π does not achieve privacy.*

Proof. It follows directly from Lemma 2 that there exists an adversary $\mathcal{A}$ with threshold $(t_a, t_p) = (0, n-1)$ corrupting a set of $(n-1)$ parties passively, say $\mathcal{E} = \{P_1, \dots P_{n-1}\}$, that is able to compute the output at the end of Round 1 itself. Thus, $\mathcal{A}$ can obtain multiple evaluations of the function f by locally plugging in different values for $\{x_1, \dots, x_{n-1}\}$ while honest P_n's input x_n remains fixed. This residual function attack 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 $x_1 \wedge x_n$). 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 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. □

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 [5]. 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 [34] 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 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 [32] in the full version [33]. 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 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 enforce 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 4. $\mathsf{LRec}^{\alpha,\beta}$ *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}.*

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

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

Proof.

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. $\mathsf{LRec}^{\alpha,\beta}$ 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 [5] that builds on an MPC protocol with abort security 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 [5] are $n-2$ apart, shooting the round complexity of reconstruction to $n-1$. We depart from the construction of [5] 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 [5]. Second, we plug in the round-optimal (2-round) MPC protocol of [10, 11] 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 purely passive parties, can. This unique summand, being the 'soft spot' for the adversary, forces him to co-operate 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 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. Our fair protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ tolerating dynamic corruption appears in Fig 3. Assumption wise, $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ relies on 2-round maliciously-secure OT in the common random/reference string model (when π_{ua} is instantiated with protocols of [10, 11]) and eNICOM (used in $\mathsf{LRec}^{\alpha,\beta}()$ and instantiated using Pedersen's commitment scheme).

Table 1: 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	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$	...	...	...	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$
$(1, n-2)$	$n-1$	$(\mathtt{N},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$	...	...	...	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$
$(2, n-3)$	$n-2$	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$	...	...	...	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$
...	...	...	...	...	...	...	...	...	...
$(i, n-i-1)$	$n-i$	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{N})$	...	$(\mathtt{N},\mathtt{Y})$	...	$(\mathtt{Y},\mathtt{Y})$	$(\mathtt{Y},\mathtt{Y})$
...	...	...	...	...	...	...	...	...	...
$(\lceil n/2\rceil - 1, \lfloor n/2\rfloor)$	$\lfloor n/2\rfloor + 1$	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{N})$	...	...	...	$(\mathtt{N},\mathtt{N})$	$(\mathtt{N},\mathtt{Y})$

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

Inputs: Party P_j has x_j for $j \in [n]$
Building blocks: (a) Protocol π_{ua} achieving security with unanimous abort against malicious majority (b) Protocol $\mathsf{LRec}^{\alpha,\beta}$ for reconstructing a (α,β)-levelled-shared value (Fig. 2); (c) Function $f_{\mathsf{LSh}}^{n-2,\lfloor\frac{n}{2}\rfloor}$ (Fig 1).
Output: $y = f(x_1 \ldots x_n)$ or $\perp$

Round $1-2$: Every P_j runs protocol π_{ua} to compute the function $f_{\mathsf{LSh}}^{n-2,\lfloor\frac{n}{2}\rfloor} \diamond f$ with input x_j to 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. 3: Fair MPC against dynamic-admissible adversary

We state the formal theorem below.

Theorem 2. *Assuming the presence of a 2-round MPC protocol π_{ua} achieving unanimous abort against malicious majority, protocol $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ with n parties satisfies correctness, achieves fairness and has a round complexity of $\lceil n/2\rceil + 1$ rounds.*

Proof. Correctness of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ follows directly from correctness of π_{ua} and $\mathsf{LRec}^{n-2,\lfloor\frac{n}{2}\rfloor}$ (Lemma 4). The security proof appears in the full version [33].

Round complexity of $\pi_{\mathsf{fair}}^{\mathsf{dyn}}$ includes 2 rounds of π_{ua} and the round complexity of $\mathsf{LRec}^{n-2,\lfloor \frac{n}{2} \rfloor}$ which is $\left(n - 2 - \lfloor \frac{n}{2} \rfloor + 1\right) = \lceil n/2 \rceil - 1$ (Lemma 4); totalling upto $\lceil n/2 \rceil + 1$ rounds. □

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 [35, 36], 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. Slapping NIZK on the 2-round broadcast-only construction of [10] readily equips it with identifiability, without inflating the round complexity. Second, we leverage the *function-delayed* property of a modified variant of the protocol of [10] (proposed by [13]) 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 failure happens to compute a new function each time as follows– (a) it hard-cores 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 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 corruptions. 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 4. Assumption wise, $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ relies on 2-round maliciously-secure OT in the common random/reference string model, NIZK (when π_{idua} is instantiated with function-delayed variant of the protocol of [10] satisfying identifiability) and eNICOM (instantiated using Pedersen's commitment scheme).

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

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

Lemma 5. *$\pi_{\mathsf{god}}^{\mathsf{dyn}}$ 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 atmost $\lceil\frac{n}{2}\rceil + 1$ (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 4 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 $\mathfrak{n} \leq \ell$.

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

Inputs: Party P_i has x_i for $i \in [n]$
Building blocks: (a) Protocol π_{idua} achieving identifiable abort 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 hardcoded 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. 4: Robust MPC against dynamic-admissible adversary

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 following number of rounds– 1 (for Round 2 of π_{idua}) + number of rounds in $\mathsf{LRec}^{\mathfrak{n}-2,\lfloor\frac{\mathfrak{n}}{2}\rfloor}$ i.e $(\mathfrak{n} - 2 - \lfloor\frac{\mathfrak{n}}{2}\rfloor + 1) = \lceil\frac{\mathfrak{n}}{2}\rceil = \lceil(\ell + 1)/2\rceil$ in total.
- 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 4 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 rerun 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 $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ 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} achieving identifiable abort against malicious majority and having function-delayed property; protocol $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ with n parties satisfies correctness, achieves guaranteed output delivery and has a round-complexity of $\lceil n/2 \rceil + 1$ rounds.*

Proof. Correctness of $\pi_{\mathsf{god}}^{\mathsf{dyn}}$ follows directly from correctness of π_{idua} and correctness of $\mathsf{LRec}^{\mathfrak{n}-2, \lfloor \frac{\mathfrak{n}}{2} \rfloor}$ (Lemma 4). The formal security proof appears in the full version [33]. Round complexity follows from Lemma 5. □

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. Recall that such an 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 [7] and passive-majority [10, 11] 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-admissble 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.

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 parametes $(t_a, t_p) = (2, 2)$ 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.

At a high level, we discuss three adversarial 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. 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 Σ_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 Σ_2 involving actively corrupt $\{P_1, P_2\}$ is designed such that corrupt P_2 obtains the same view in Σ_2 as an honest P_2 in Σ_1 and therefore computes the output $\widetilde{y}$ at the end of Σ_2. (Here, view of P_i includes x_i, r_i, the messages received during π and the knowledge related to CRS and PKI setup.) 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 Σ_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.

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^ℓ_i denotes the view of party P_i at the end of execution Σ_ℓ ($\ell \in [3]$) of π. Below we describe the strategies $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$.

$\mathcal{A}_1$: $\mathcal{A}$ 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 now present a sequence of lemmas to complete the proof.

Lemma 6. *At the end of Σ_1, parties compute output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$, 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 correctness and robustness of π 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' 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. Then, in an execution of π, an adversary corrupting $\{P_2, P_3, P_4, P_5\}$ purely passively (correponding to $(t_a, t_p) = (0, 4)$) can learn the output 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 in the full version [33]). 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. □

Lemma 7. *At the end of Σ_2, parties compute output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$, 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 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. It now follows directly from Lemma 6 that P_2 computes the output $\widetilde{y} = f(\widetilde{x_1}, x_2, x_3, x_4, x_5)$. By correctness and robustness of π computing the common output function f, it must hold that all parties output $\widetilde{y}$ at the end of Σ_2. □

Lemma 8. *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 $\widetilde{y}$.*

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 $\widetilde{y}$ (Lemma 7). □

Lemma 9. *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.*

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 since $\mathcal{A}_3$ involves only semi-honest corruptions. Next, it is easy to check 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 8 that the adversary can obtain the output $\widetilde{y}$ as well.

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 $\widetilde{y}$ which violates the security of π. (We give a concrete example of such an f that breaches privacy of honest P_2 in the full version.) This completes the proof of the lemma. □

Thus, we have arrived at a contradiction to our assumption that π is secure; completing the proof of Theorem 4. □

We present a natural extension of the above proof for $n > 5$, a concrete example of f and a brief intuition of why the above lower bound argument does not hold when malicious corruption $t_a \leq 1$ in the full version [33].

5.2 Impossibility of 2-round Fair MPC

We begin with the observation that the existing 3-round lower bounds of [6, 7, 8] for fair malicious-minority MPC do not carry over in our setting. The lower bound of both [6, 7] break down when the parties have access to a PKI (as acknowledged/demonstrated in their work). The result of [8], assuming access to pairwise-private and broadcast channels, also breaks down when parties have access to a PKI (elaborated in the full version [33]). The proof, originally given without the mention of CRS, seems to withstand a CRS.

We now present our lower bound formally.

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 a 2-round 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 incase of no active corruptions i.e $(t_a, t_p) = (0, 2)$). On a high-level, 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 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. We now present a sequence of claims to complete the formal proof.

Lemma 10. *Protocol π must be such that the combined view of $\{P_2, P_3\}$ at the end of Round 1 suffices for output computation.*

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 due to correctness of π, as he receives the entire protocol communication as per honest execution. Since π is fair, the honest parties $\{P_2, P_3\}$ must also obtain the output at the end of π; even 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. □

Lemma 11. *There exists an adversarial strategy such that the adversary obtains the output at the end of Round 1.*

Proof. The proof follows directly from Lemma 10– $\mathcal{A}$ corrupting $\{P_2, P_3\}$ purely passively ($(t_a, t_p) = (0, 2)$) would obtain the output at the end of Round 1. □

Lemma 12. *Protocol π does not achieve privacy.*

Proof. It is implied from Lemma 11 that $\mathcal{A}$ corrupting $\{P_2, P_3\}$ purely passively can obtain multiple evaluations of the function f 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 3 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. Our fair 3-round upper bound can be arrived at by simplifying the robust general-case construction and appears in full version [33]. 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 [37, 28] 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. 5.*

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

1. α *shamir-sharing of secret v:* Choose random $a_1, a_2 \ldots a_\alpha \in \mathbb{F}$, where $\mathbb{F}$ denotes a finite field. Build the α-degree polynomial $A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{\alpha-1}x^{\alpha-1} + a_\alpha x^\alpha$, 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) [38] 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. 5: 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 6) amongst the parties. We state the relevant properties below (proof appears in the full version [33]):

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. 6: Protocol for Reconstruction of an authenticated-secret

Lemma 13. *The pair $(f^{\alpha}_{\mathsf{ASh}}, \mathsf{ARec}^{\alpha})$ satisfies the following:*

i. Privacy. *For all $v \in \mathbb{F}$, the output $(S_1, \ldots, S_n) \leftarrow f^{\alpha}_{\mathsf{ASh}}(v)$ satisfies the following– $\forall \{i_1, \ldots i_{\alpha'}\} \subset [n]$ with $\alpha' \leq \alpha$, the distribution of $\{S_{i_1}, \ldots, S_{i_{\alpha'}}\}$ 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 atleast $\alpha + 1$ 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.*

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, [28] 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 honest-majority (malicious minority) 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 [28]. 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 [28]. First, we invoke a 2-round actively-secure protocol π_{idua} with identifiable abort against malicious majority (can be instantiated with protocols of [10, 11] augmented with NIZKs) 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 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) additional 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 [7, 13] allows us to superimpose this round with the run of π_{idua}. Both these instantations 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). Our protocol appears in Fig. 7. Assumption wise, $\pi^{\mathsf{bou}}_{\mathsf{god}}$ relies on 2-round maliciously-secure OT in the common random/reference string model, NIZK (when π_{idua} is instantiated with function-delayed variant of the protocol of [10] satisfying identifiability), Zaps and public-key encryption (when π_{god} is instantiated with protocol of [13]).

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

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; (b) 3-round honest-majority actively-secure robust protocol π_{god} with additional property of Round 1 being function and input independent; (c) Protocol $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ for reconstructing an $\lfloor n/2 \rfloor$-authenticated-shared secret (Fig. 6); (d) Function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}$ (Fig. 5).

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

Round 1–2: The parties run π_{idua} computing the function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f$ with input x_i to obtain output $(S_i = (\mathsf{sh}_i, \{k_{ji}\}_{j \in [n]}, \{\mathsf{tag}_{ij}\}_{j \in [n]}), \mathcal{B})$, 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 $S_i = \perp$, 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. 7: Robust MPC against boundary-admissible adversary

We state the formal theorem below.

Theorem 6. *Assuming the presence of a 2-round protocol π_{idua} achieving identifiable abort against malicious majority and a 3-round robust protocol π_{god} against malicious minority (with special property of Round 1 being function and input-independent), the 4-round MPC protocol $\pi_{\mathsf{god}}^{\mathsf{bou}}$ (Figure 7) satisfies correctness and achieves guaranteed output delivery against boundary-admissible $\mathcal{A}$.*

Proof. Correctness of $\pi_{\mathsf{god}}^{\mathsf{bou}}$ follows directly from that of π_{idua}, π_{god} and $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ (Lemma 13). We prove its security in the full version [33]. □

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} with Round 1 of π_{idua} can be avoided and the 2-round robust protocol of [7] against malicious minority 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 7) 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 atmost 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 [10] to instantiate π_{idua} that is function-delayed and achieves identifiability. The formal description of $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ appears in Fig 8. This upper bound 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 and NIZK (when π_{idua} is instantiated with above mentioned variant of the protocol of [10]).

Protocol $\pi_{\mathsf{god}}^{\mathsf{bou},1}$

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 and having function-delayed property; (b) Protocol $\mathsf{ARec}^{\lfloor n/2 \rfloor}$ for reconstructing an $\lfloor n/2 \rfloor$-authenticated-shared secret (Fig. 6); (c) Function $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor}$ (Fig. 5).

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 $f_{\mathsf{ASh}}^{\lfloor n/2 \rfloor} \diamond f$ 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 hardcoded 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. 8: Robust MPC against special-case boundary-admissible adversary

We state the formal theorem below.

Theorem 7. *Assuming the presence of a 2-round protocol π_{idua} achieving identifiable abort against malicious majority and having function-delayed property, the 3-round MPC protocol $\pi_{\mathsf{god}}^{\mathsf{bou},1}$ (Figure 8) satisfies correctness and achieves guaranteed output delivery 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 13). We prove its security in full version [33]. □

References

1. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or A completeness theorem for protocols with honest majority," in *ACM STOC*, 1987.
2. D. Chaum, I. Damgård, and J. Graaf, "Multiparty computations ensuring privacy of each party's input and correctness of the result," in *CRYPTO*, 1987.
3. A. C. Yao, "Protocols for secure computations (extended abstract)," in *FOCS*, 1982.
4. R. Cleve, "Limits on the security of coin flips when half the processors are faulty (extended abstract)," in *ACM STOC*, 1986.
5. M. Hirt, C. Lucas, and U. Maurer, "A dynamic tradeoff between active and passive corruptions in secure multi-party computation," in *CRYPTO*, 2013.
6. R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin, "On 2-round secure multiparty computation," in *CRYPTO*, 2002.
7. S. D. Gordon, F. Liu, and E. Shi, "Constant-round MPC with fairness and guarantee of output delivery," in *CRYPTO*, 2015.
8. A. Patra and D. Ravi, "On the exact round complexity of secure three-party computation," in *CRYPTO*, 2018.
9. S. Halevi, Y. Lindell, and B. Pinkas, "Secure computation on the web: Computing without simultaneous interaction," in *CRYPTO*, 2011.
10. S. Garg and A. Srinivasan, "Two-round multiparty secure computation from minimal assumptions," in *EUROCRYPT*, 2018.
11. F. Benhamouda and H. Lin, "k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits," in *EUROCRYPT*, 2018.
12. T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority (extended abstract)," in *STOC*, 1989.
13. P. Ananth, A. R. Choudhuri, A. Goel, and A. Jain, "Round-optimal secure multiparty computation with honest majority," in *CRYPTO*, 2018.
14. S. Badrinarayanan, A. Jain, N. Manohar, and A. Sahai, "Secure MPC: laziness leads to GOD," *IACR Cryptology ePrint Archive*, vol. 2018, p. 580, 2018.
15. B. Applebaum, Z. Brakerski, and R. Tsabary, "Degree 2 is complete for the round-complexity of malicious MPC," in *EUROCRYPT*, 2019.
16. Y. Ishai, R. Kumaresan, E. Kushilevitz, and A. Paskin-Cherniavsky, "Secure computation with minimal interaction, revisited," in *CRYPTO*, 2015.
17. D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly secure message transmission," *J. ACM*, vol. 40, no. 1, pp. 17–47, 1993.
18. M. Fitzi, M. Hirt, and U. M. Maurer, "Trading correctness for privacy in unconditional multi-party computation (extended abstract)," in *CRYPTO*, 1998.
19. M. Fitzi, M. Hirt, and U. M. Maurer, "General adversaries in unconditional multiparty computation," in *ASIACRYPT*, 1999.

20. M. Hirt, U. M. Maurer, and V. Zikas, “MPC vs. SFE : Unconditional and computational security,” in *ASIACRYPT*, 2008.
21. Z. Beerliová-Trubíniová, M. Fitzi, M. Hirt, U. M. Maurer, and V. Zikas, “MPC vs. SFE: perfect security in a unified corruption model,” in *TCC*, 2008.
22. D. Chaum, “The spymasters double-agent problem: Multiparty computations secure unconditionally from minorities and cryptographically from majorities,” in *CRYPTO*, 1989.
23. C. Lucas, D. Raub, and U. M. Maurer, “Hybrid-secure MPC: trading information-theoretic robustness for computational privacy,” in *ACM PODC*, 2010.
24. M. Fitzi, M. Hirt, T. Holenstein, and J. Wullschleger, “Two-threshold broadcast and detectable multi-party computation,” in *EUROCRYPT*, 2003.
25. M. Fitzi, T. Holenstein, and J. Wullschleger, “Multi-party computation with hybrid security,” in *EUROCRYPT*, 2004.
26. Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank, “On combining privacy with guaranteed output delivery in secure multiparty computation,” in *CRYPTO*, 2006.
27. J. Katz, “On achieving the "best of both worlds" in secure multiparty computation,” in *ACM STOC*, 2007.
28. Y. Ishai, J. Katz, E. Kushilevitz, Y. Lindell, and E. Petrank, “On achieving the "best of both worlds" in secure multiparty computation,” *SIAM J. Comput.*, vol. 40, no. 1, 2011.
29. M. Hirt, C. Lucas, U. Maurer, and D. Raub, “Graceful degradation in multi-party computation (extended abstract),” in *ICITS*, 2011.
30. A. Patra, A. Choudhary, T. Rabin, and C. P. Rangan, “The round complexity of verifiable secret sharing revisited,” in *CRYPTO*, 2009.
31. M. Backes, A. Kate, and A. Patra, “Computational verifiable secret sharing revisited,” in *ASIACRYPT*, 2011.
32. T. P. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” in *CRYPTO*, 1991.
33. A. Patra and D. Ravi, “Beyond honest majority: The round complexity of fair and robust multi-party computation.” Cryptology ePrint Archive, Report 2019/998, 2019. https://eprint.iacr.org/2019/998.
34. A. Shamir, “How to share a secret,” *Commun. ACM*, vol. 22, 1979.
35. M. Hirt, U. M. Maurer, and B. Przydatek, “Efficient secure multi-party computation,” in *ASIACRYPT*, 2000.
36. M. Hirt and U. M. Maurer, “Robustness for free in unconditional multi-party computation,” in *CRYPTO*, 2001.
37. Y. Ishai, E. Kushilevitz, M. Prabhakaran, A. Sahai, and C. Yu, “Secure protocol transformations,” in *CRYPTO*, 2016.
38. O. Goldreich, *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.