Information-Theoretic Broadcast-Optimal MPC

Michele Ciampi[1*], Ivan Damgård[2], Divya Ravi[3†], Luisa Siniscalchi[4], and Sophia Yakoubov[2]

[1] The University of Edinburgh, UK; michele.ciampi@ed.ac.uk
[2] Aarhus University, Denmark; {ivan, sophia.yakoubov}@cs.au.dk
[3] University of Amsterdam, Netherlands; d.ravi@uva.nl
[4] The Technical University of Denmark, Denmark; luisi@dtu.dk

Abstract. Broadcast, though often used as a black box in cryptographic protocols, is expensive to realize in terms of rounds and communication complexity. We investigate the minimal use of broadcast in round-optimal information-theoretic MPC, with statistical security. For information-theoretic MPC with guaranteed output delivery, four rounds of communication are necessary and sufficient (Applebaum, Kachlon and Patra, FOCS 2020; Applebaum, Kachlon and Patra, STOC 2023). We show that broadcast is unavoidable in the second and third rounds of statistical MPC protocols. To complement our lower bounds, we modify the protocol of Applebaum, Kachlon and Patra (STOC 2023) to make use of broadcast only in the second and third round. Along the way, we show that the sharing phase of any three-round information-theoretic VSS protocol must also make use of broadcast in the second and third rounds.

1 Introduction

Round complexity is one of the most important efficiency metrics of interactive protocols. Each round of communication can take a long time, especially if the protocol runs over a high-latency network. Finding *round-optimal* protocols has been an important area of study within secure multi-party computation (MPC), where a set of parties want to compute a function on private held inputs such that the intended result is the only new information released. Two rounds have been shown to be sufficient and necessary for computational MPC [HLP11,GGHR14]; four rounds have been shown to be sufficient and necessary for information-theoretic MPC with optimal resilience [AKP20b,AKP20a,AKP23].

Towards designing efficient protocols, it is also important to use *reliable broadcast* in as few rounds as possible. Most MPC protocols make liberal use of broadcast, but in practice, broadcast requires either multiple rounds of point-to-point communication, or the use of costly resources such as physical channels or an external service such as a blockchain.

*Partially supported by the Input Output Research Hub (IORH) of the University of Edinburgh and by Sunday Group Inc.

†Funded by the gravitation project Challenges in Cyber Security (CiCS) with file number 024.006.037 which is financed by the Dutch Research Council (NWO).

An MPC protocol that makes minimal use of broadcast is said to be *broadcast-optimal*; this notion has been studied before in many different settings, as detailed in the section on related work (Section 1.1). The case of broadcast-optimal information-theoretic MPC has not yet been studied. In this paper we initiate the study of *broadcast-optimal information-theoretic MPC*; that is, we determine which of the four rounds necessary for information-theoretic MPC truly must make use of broadcast.

Our Contributions We focus on MPC that achieves guaranteed output delivery with statistical security and optimal corruption threshold $t < \frac{n}{2}$, where n is the number of parties. We show that the four-round protocol of Applebaum, Kachlon and Patra [AKP23], which uses broadcast in every round except the first, remains secure if broadcast is removed from the last round as well. We then show that a protocol without broadcast in the second round cannot be secure; similarly, we show that a protocol without broadcast in the third round cannot be secure, proving the broadcast-optimality of the (modified) protocol of Applebaum *et al.*[5]

Along the way, we study the broadcast complexity of verifiable secret sharing (VSS). VSS has two phases: a sharing phase, at the end of which the dealer's message must be committed, and a reconstruction phase. It is known that three rounds of communication are necessary and sufficient for the sharing phase of VSS [PCRR09,KPR10,AKP23]. We show that broadcast is necessary in the second and third of these rounds in the sharing phase, just as in general-purpose MPC. This holds regardless of the number of rounds used in the reconstruction phase.

Note that the VSS from [AKP23] has three-round sharing and one-round reconstruction, so this VSS is broadcast optimal given our results: it uses broadcast only in the second and third round of sharing, and although it was described as using broadcast during reconstruction, we observe that in any VSS with one round reconstruction, that round can run on point-to-point channels.

1.1 Related Work

There are two lines of work which relate to this paper: *broadcast-optimal MPC*, and *round-optimal information-theoretic MPC and VSS*.

Broadcast-Optimal MPC Over the past five years, broadcast-optimal MPC has been studied in a number of settings. Cohen, Garay and Zikas [CGZ20] introduced the topic of broadcast-optimal MPC, and gave a complete characterization of which rounds require broadcast for different security guarantees in the computational, dishonest majority setting. Damgård *et al.* [DMR+21] extended the study of broadcast-optimal MPC to the computational, honest majority setting with a PKI; Damgård *et al.* [DRSY22] then completed the picture for the computational, honest majority setting without a PKI. Damgård *et al.* [DRSY23]

[5] Throughout, we consider security with respect to straight-line simulation, indeed all known information theoretic protocols have this type of security.

studied what happens when the assumption of synchrony is relaxed; and finally, Ciampi *et al.* [CDR+23] characterized four-round broadcast-optimal MPC in the computational, dishonest majority setting with no setup at all (in the plain model). Our paper is the first to study broadcast-optimality for information-theoretic protocols.

Round-Optimal Information-Theoretic VSS and MPC Gennaro *et al.* [GIKR01] showed that two rounds are sufficient and necessary for perfectly secure VSS with $t < \frac{n}{4}$, and that three rounds are sufficient and necessary for perfectly secure VSS with $t < \frac{n}{3}$. Applebaum *et al.* [AKP20b] showed that four rounds are sufficient and necessary for perfectly secure MPC with $t < \frac{n}{3}$. Patra *et al.* [PCRR09] showed that three rounds are necessary for statistically secure VSS, which was shown to be sufficient in [KPR10,AKP23]. Finally, Applebaum *et al.* showed that four rounds are necessary [AKP20a] and sufficient [AKP23] for statistically secure MPC with $t < \frac{n}{2}$, which is the setting that we study.

Broadcast-Efficient VSS and MPC Garay *et al.* [GGOR14] construct a constant-round statistical VSS with $t < \frac{n}{2}$ that uses broadcast in only two of the rounds. However, their overall number of rounds is twenty. In the eprint version [GGOR12], they extend this to construct a constant-round MPC which uses only three broadcast rounds in the same setting, and prove that two broadcast rounds are necessary.

1.2 Model and Notation

We consider protocols executed over a synchronous network, featuring private point-to-point channels and an authenticated broadcast channel. Computation proceeds in rounds, where in each round, parties can send private messages to specific recipients and broadcast messages to all other parties. Importantly, the adversary cannot read or alter messages transmitted over the point-to-point channels, and the broadcast channel is authenticated, ensuring that all parties can verify the sender's identity and preventing any tampering by the adversary.

We assume that parties start out with no correlated randomness. This is especially important for our VSS lower bounds: given correlated randomness comprising a VSS of a random value (where the random value is also given to the dealer), VSS becomes trivial. All the dealer would need to do to share a secret is to broadcast the secret masked with the random value.

When talking about broadcast patterns, we use a shorthand, with `bc` representing a broadcast round (in which point-to-point channels are also available) and `p2p` representing a round where only point-to-point channels are available. So, a `bc-bc-p2p` protocol would be a three-round protocol where broadcast is not available in the final round. Finally, we use the notation $[a, b]$ to denote the set of indices $\{a, a+1, \ldots, b\}$ and $[a]$ to denote the set of indices $\{1, 2, \ldots, a\}$.

1.3 Results

Our results for verifiable secret sharing (VSS) are summarized in Table 1. As long as $n \leq 2.5t$, our lower bounds indicate that the $\texttt{p2p-bc}^2$ VSS construction of Applebaum *et al.* [AKP23] has an optimal three-round sharing phase in terms of broadcast usage. Furthermore, their reconstruction phase is only one round, and in this case, broadcast is not needed for reconstruction (details appear in our full version, [CDR+25]). Thus, this VSS is broadcast optimally.

The feasibility of `bc-bc-p2p` VSS when $n > 2.5t$ remains an open problem.

broadcast pattern	**security**	**threshold**	**feasible?**
$\texttt{p2p-bc}^2$	statistical	$n = 2t + 1$	✓([AKP23])
$\texttt{p2p}^{(R-1)}\texttt{-bc}$ for any R	computational	$n \leq 3t$	✗(Theorem 1)
$\texttt{bc-p2p}^{(R-2)}\texttt{-bc}$ for any R	statistical	$n \leq 3t$	✗(Theorem 2)
`bc-bc-p2p`	statistical	$n \leq 2.5t$	✗(Theorem 5)

Table 1: Summary of Results for Verifiable Secret Sharing

Our results for MPC are summarized in Table 2. In addition to our lower bounds, in Theorem 8 we claim that under certain conditions, the last round of an MPC protocol can be run over point-to-point channels without compromising security. Our lower bounds, together with this observation, indicate that the MPC construction of Applebaum *et al.* [AKP23] is an optimal four-round construction in terms of broadcast usage as long as $n \leq 2.5t$. The case of $n > 2.5t$ remains an open problem.

Arguably, the most interesting cases are when t is maximal such that VSS/MPC is possible, i.e., $n = 2t+1$. Our results hold for all such cases except $n = 3, t = 1$[6].

broadcast pattern	**security**	**threshold**	**other restrictions**	**feasible?**
`p2p-bc-bc-p2p`	statistical	$n = 2t + 1$	N/A	✓([AKP23], Theorem 8)
`bc-p2p-bc-bc`	statistical	$n \leq 3t$	no setup, straight-line simulation	✗(Theorem 3)
`bc-bc-p2p-bc`	statistical	$n \leq 2.5t$	no setup, straight-line simulation	✗(Theorem 7)

Table 2: Summary of Results for MPC

[6] for even n, they hold for $n \geq 10$.

1.4 Technical Overview

The Necessity of Broadcast in the Second Round In Theorem 1, we claim that it is impossible to build VSS for $t \geq \frac{n}{3}$ with broadcast only in the last round, even if it is preceded by an arbitrary number of point-to-point rounds, and even if only *computational* security is required. We prove this by studying the $n = 3, t = 1$ case and assuming there exists a secure VSS for this setting. We call our three parties P_1, P_2 and P_3, where P_1 always plays the role of the dealer. For the proof we introduce a "thought experiment", a game played by 5 parties. Namely we introduce two versions of each of P_2 and P_3, so we now have P_{2a}, P_{2b}, P_{3a} and P_{3b}. All parties are programmed to execute the protocol honestly. When executing the thought experiment, we let all five parties run the protocol, but we block some of the communication, such that the only pairs of parties who get to communicate will be $(P_1, P_{2b}), (P_1, P_{3a}), (P_{2a}, P_{3a})$ and (P_{2b}, P_{3b}). Our attack on the protocol will corrupt one of P_1, P_{2a} and P_{3a}, and the corrupt party will run two of our five parties in her head while interacting in real life with the remaining two, according to the way the thought experiment was set up. Specifically, corrupt P_1, imagining P_{2b} and P_{3b} in her head, will interact with P_{3a} but will not talk to P_{2a}. Similarly, corrupt P_{2a} imagines P_{2b} and P_{3b} in her head, while corrupt P_{3a} imagines P_{2a} and P_{3b}. In all three attacks, we are actually executing the same five-player game, so therefore the view of each of our five parties is identically distributed in each attack, whether that party is imagined or not [7]. We illustrate the three attacks in Figure 1.

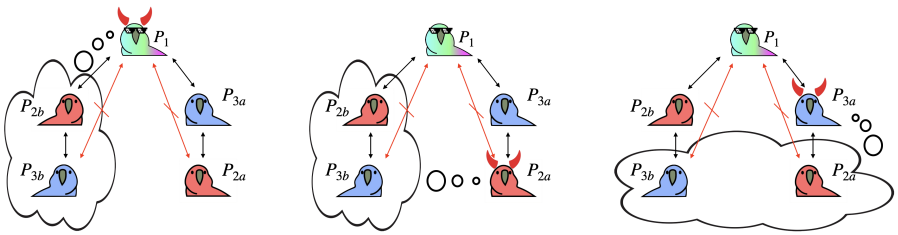


Fig. 1: Scenarios considered in the proof of Theorem 1

[7] Note that there is in fact a difference between the attacks because the messages that are physically broadcast are only those coming from real parties, not imagined ones. So, in the first and second attack, we will see broadcasts from P_1, P_{2a} and P_{3a}, but in the third attack from P_1, P_{2b} and P_{3a}. We are saved by the fact that there is only broadcast in the last round, so there is no time to respond to the broadcast messages. Therefore, the view of any of our 5 parties is indeed identically distributed, whether that party is imagined or not.

We then argue as follows: The commitment property of VSS says that the correct secret must be reconstructed, whether or not the dealer participates. We use this and the first two attacks to show that P_{2a} and P_{3a} must be able to reconstruct the secret on their own. But this implies that in the third attack, corrupt P_{3a} has enough information to reconstruct the secret on her own, violating privacy.

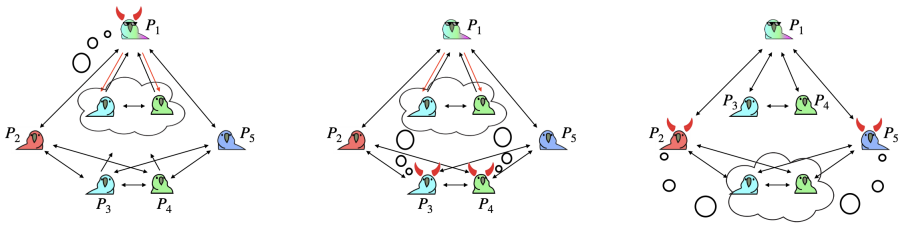
In Theorem 2 we claim that, if information theoretic security is desired, an additional round of broadcast at the beginning of the VSS sharing phase does not help. This is true since parties are able to sample randomness for one another which would be consistent with the initial broadcast message, and are thus still able to carry out the strategies described above, wherein they run fictitious versions of one-another.

Finally, in Theorem 3 we claim that MPC with information theoretic security is unattainable in four rounds without broadcast in the second round. We prove this via a sequence of adversarial strategies closely related to those we used to show the corresponding result for VSS.

The Necessity of Broadcast in the Third Round In Theorem 5, we show that 3-round VSS without broadcast in the last round is impossible for $2t < n \leq 2.5t$, by first studying the case of $n = 5$ and $t = 2$ and assuming (for contradiction) that there exists a secure VSS in this setting. P_1 will be the dealer throughout.

At a very high level, we first consider two scenarios: In Scenario 1, P_1 sends nothing to P_3 and P_4. In addition, she pretends to have received first-round point-to-point messages computed using a different randomness on behalf of P_3 and P_4 (in a way that is still consistent with their first-round broadcast). In Scenario 2, however, P_3 and P_4 are corrupt instead. They use a different randomness for their communication with P_1, and pretend they heard nothing from P_1. Further, the corrupt parties in both scenarios behaves honestly only towards P_2 and P_5 in the third round. From the perspective of honest P_2 and P_5, the two scenarios are indistinguishable: the absence of a broadcast in the third round makes any kind of conflict resolution impossible. Scenario 2 involves an honest dealer and must therefore lead to reconstruction of its secret; so must Scenario 1. Intuitively, the only information about the secret that P_1 distributed during the sharing phase in Scenario 1 is available to P_2 and P_5 (as no relevant information was privately communicated to P_3 and P_4). This leads us to a contradiction as we show that this can be exploited to breach privacy in Scenario 3, where P_2 and P_5 are corrupt. We illustrate the three scenarios in Figure 2.

The proof for the corresponding result on MPC (Theorem 7) follows a very similar strategy, except that here, we need to carefully select a function f for which we can show that computing it without broadcast in the third round is impossible. At a very high-level, similar to the VSS proof, we design a scenario with a corrupt P_1, where the views of the honest parties at the end of the first three rounds does not contain any information about P_1's input. However, the scenario and chosen function is designed such that the adversary is still



Scenario 1 **Scenario 2** **Scenario 3**

Fig. 2: Scenarios considered in the proof of Theorem 5

able to obtain the output via its corrupt friend (say P_2) who behaved honestly throughout. Therefore, to prove security against such an adversary, there must exist a straight-line simulator that is able to extract P_1's input at the end of the first three rounds (as the adversary could be rushing in the last round). However, this is impossible as the honest parties' views are independent of P_1's input, leading us to a contradiction.

2 Definitions

2.1 Verifiable Secret Sharing (VSS)

The following definition is taken almost verbatim from [AKP23].

Definition 1 (Statistical (resp. Computational) Secure VSS). *Let Y be a finite domain, $|Y| \geq 2$, and let $\mathcal{P}$ be a set of parties that includes a distinguished dealer $D \in \mathcal{P}$. A VSS protocol consists of two phases, a sharing phase and a reconstruction phase, with the following syntax.*

- *Sharing: At the beginning, D holds a secret $x \in Y$ and each party including the dealer holds an independent random input r_i. The sharing phase may span over several rounds. In each round, each party can privately send messages to the other parties and it can also broadcast a message. Each message sent or broadcast by P_i is determined by the view of P_i, consists of its input (if any), its random input and messages received from other parties in previous rounds. Specifically, an n-party R-round sharing phase consists of a set of next-message functions $\{\mathtt{sh}_i^{(\rho)}\}_{i\in[n],\rho\in R}$. $\mathtt{sh}_i^{(\rho)}$ denotes the algorithm executed by P_i in round ρ of the sharing phase, where $i \in [n], \rho \in [R]$. $\mathtt{sh}_i^{(\rho)}$ takes as input the* view *of the party, which consists of the randomness, the input, and all the messages received up to round $\rho - 1$ by P_i. The output of $\mathtt{sh}_i^{(\rho)}$ consists of the set of the point-to-point messages and broadcast messages that need to be sent in round ρ.*

– *Reconstruction: At the beginning of the reconstruction, the parties hold their view from the sharing phase. The reconstruction phase may span over several rounds, and at each round the parties send messages based on their view. At the end of the reconstruction, each party outputs a value.*

Let $\epsilon \geq 0$ (resp. $\epsilon = \mathsf{negl}(\lambda)$). A two-phase, n-party protocol as above is called an ϵ-secure (n,t)-VSS, if for any adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{sh}}, \mathcal{A}_{\mathsf{rec}})$ corrupting at most t parties, the following holds:

– *Correctness: If D is honest then all honest parties output x at the end of the reconstruction phase, with probability at least $1-\epsilon$.*
– *Privacy: If D is honest then the adversary's view during the sharing phase reveals almost no information about x. Formally, let D_x be the random variable describing the view of $\mathcal{A}$ in the sharing phase on secret x. Then, for any $x' \neq x$ the distributions of the random variables D_x and $D_{x'}$ are ϵ-close in statistical-distance (resp. computaional indistiguishable).*
– *Commitment: If D is corrupt then, except with probability $1-\epsilon$, at the end of the sharing phase there is a value $x^* \in Y$ such that at the end of the reconstruction phase the output is x^*. More formally, we assume that an adversary $\mathcal{A}$ that corrupts D is a two-phase adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{sh}}, \mathcal{A}_{\mathsf{rec}})$ where $\mathcal{A}_{\mathsf{sh}}$ takes randomness $r_{\mathcal{A}}$, plays the sharing phase and outputs a state Z. At the reconstruction phase $\mathcal{A}_{\mathsf{rec}}$ gets Z and, in addition, a bit σ and tries to flip the outcome depending on σ. Specifically, let $\mathbb{H}$ denote the set of honest parties. For $r = \{r_i\}_{i\in\mathbb{H}}$ and $\sigma \in \{0,1\}$, denote by $y_{r,r_{\mathcal{A}}}(i,\sigma)$ the final output of party P_i, with $i \in \mathbb{H}$, in an execution with $\mathcal{A}_{\mathsf{sh}}(r_{\mathcal{A}}), \mathcal{A}_{\mathsf{rec}}(Z,\sigma)$ where the random tape of an honest party P_j is set to r_j. Then the commitment property requires that*

$$\Pr_{r,r_{\mathcal{A}}}[\exists x^* \in Y : \forall i \in \mathbb{H}, \sigma \in \{0,1\}, x^* = y_{r,r_{\mathcal{A}}}, (i,\sigma)] > 1-\epsilon.$$

2.2 Secure Multi-Party Computation (MPC)

In this section, we recall the notions of computational, statistical, and perfect security for MPC. All the notions we consider are defined with respect to malicious adversaries and ensure that honest parties always get the output (i.e., guaranteed output delivery, abbreviated to G.O.D.). We consider malicious adversaries capable of employing any arbitrary strategy and static (non-adaptive) corruptions where the adversary corrupts the parties at the onset of the execution. Security is defined by comparing a real protocol execution with an ideal model, where participants simply submit their inputs to a trusted party and receive their outputs in return. For a detailed explanation of these real and ideal executions we refer to [Can00,Gol04]. Here we provide an overview following the style of [AL11].

Real model. In the real model, a set of parties $P_1, \ldots, P_n$ execute a protocol Π. An n party R-round protocol Π consists of a set of next message functions

$((\texttt{nxt-msg}_i^{(\rho)})_{i\in[n],\rho\in[R]}, (\texttt{output}_i)_{i\in[n]})$. $\texttt{nxt-msg}_i^{(\rho)}$ denotes the algorithm executed by P_i in round ρ of the MPC protocol, where $i \in [n], \rho \in [R]$. $\texttt{nxt-msg}_i^{(\rho)}$ takes as input the *view* of the party, which consists of the input, the randomness, and all the messages received up to round $\rho - 1$. The output of $\texttt{nxt-msg}_i^{(\rho)}$ consists of the set of the `p2p` and `bc` messages that need to be sent in round ρ. $\texttt{output}_i$ denotes the procedure to computes the final output of the computation for P_i. The input of $\texttt{output}_i$ consists of the view of P_i at the end of round R.

The adversary is assumed to be *rushing*, meaning that in each round, it can observe the messages sent by honest parties before deciding on the messages to be sent by the corrupt parties. Let Π be an n-party protocol, $\mathcal{A}$ be an arbitrary machine with auxiliary input z, and $I \subset [n]$ represent the set of corrupted parties controlled by $\mathcal{A}$. We define $\text{REAL}_{\Pi,\mathcal{A}(z),I}(\mathbf{x})$ as the random variable representing the adversary's view and the outputs of the honest parties during a real execution of Π in the described model, where each party P_i (for $i \in [n]$) has input x_i, with $\mathbf{x} := (x_1, \ldots, x_n)$.

Ideal model. for a function f, the parties submit their inputs to a trusted party, which then computes and returns the output. The ideal adversary, denoted by $\mathcal{S}$ (acting as a "simulator"), controls a subset of corrupted parties, represented by I. An execution in this model proceeds as follows:

- **Input Stage:** The adversary $\mathcal{S}$ receives auxiliary input z and observes the inputs x_i of the corrupted parties P_i (for all $i \in I$). $\mathcal{S}$ has the ability to replace any x_i with an alternative x_i' of its choosing, provided that $|x_i'| = |x_i|$.
- **Computation:** Each party transmits its (potentially modified) input to the trusted party. Denote the received inputs by $x_1', \ldots, x_n'$. The trusted party then computes $(y_1, \ldots, y_n) = f(x_1', \ldots, x_n')$ and sends y_j to each corresponding party P_j for $j \in [n]$.
- **Outputs:** Each honest party P_j (where $j \notin I$) outputs y_j, while corrupt parties return $\perp$. The adversary $\mathcal{S}$ may produce an arbitrary function of its observed information.

Throughout this paper, we refer to interactions between the parties and the functionality. For instance, when stating that a party submits its input to the functionality, we imply that the input is transmitted to the trusted party, which executes the computation.

We denote by $\text{IDEAL}_{f,\mathcal{S}(z),I}(\mathbf{x})$ the outputs of the ideal adversary $\mathcal{S}$ (who controls the corrupted parties in I) and the outputs of the honest parties after an ideal execution with a trusted party computing f, given inputs $x_1, \ldots, x_n$ from the parties and auxiliary input z from $\mathcal{S}$. Notably, communication between the trusted party and the participants $P_1, \ldots, P_n$ occurs over an ideal private channel.

Definition of Security. Informally, a protocol is considered secure if its real-world execution can be simulated within the ideal model. Specifically, for every adversary $\mathcal{A}$ attacking the real model, there must exist an adversary $\mathcal{S}$ in the

ideal model such that the distribution of outputs if the real execution of the protocol with $\mathcal{A}$ matches that of an ideal execution with $\mathcal{S}$. This ensures that any power $\mathcal{A}$ has in a real execution is equivalent to what $\mathcal{S}$ can achieve in the ideal model.

In defining security, we impose the requirement that the ideal-model adversary $\mathcal{S}$ runs in polynomial time relative to the running time of $\mathcal{A}$, regardless of how long $\mathcal{A}$ takes to execute. As discussed in [Can00,Gol04], this requirement is crucial because it ensures that information-theoretic security implies computational security. When this condition holds, we say that $\mathcal{S}$ has comparable complexity to $\mathcal{A}$.

Definition 2 (Security for MPC). *Let $f : (0,1^*)^n \to (0,1^*)^n$ be an n-ary functionality, and let Π be a protocol. We say that Π is secure against t corruptions if there exists a probabilistic simulator $\mathcal{S}$, such that for every probabilistic unbounded adversary $\mathcal{A}$ in the real model, for every subset $I \subset [n]$ with $|I| \leq t$, every input vector $\mathbf{x} \in (0,1^*)^n$ where $|x_1| = \cdots = |x_n|$, and every auxiliary input $z \in \{0,1\}^*$ it holds that the following are statistically indistinguishable, where $\mathcal{S}$ has black-box access to $\mathcal{A}$ and runs in time comparable to $\mathcal{A}$:*

$$\{IDEAL_{f,\mathcal{S}(z),I}(\mathbf{x})\}, \{REAL_{\Pi,\mathcal{A}(z),I}(\mathbf{x})\}.$$

We consider *straight-line* simulation, where the simulator does not rewind $\mathcal{A}$.

3 On the Necessity of Broadcast in the Second Round

In this section we argue that statistically-secure three-round VSS, and statistically-secure four-round MPC, is infeasible without broadcast in the second round. We formally demonstrate the validity of these claims by first proving a more generic impossibility result that holds for $R \geq 4$-round protocols. In Section 3.1, we start by showing that even given computational assumptions, no R-round VSS can exist if broadcast is not available until the last round. In Section 3.2, we show that without computational assumptions, R-round VSS is impossible if broadcast is available only on the first and the last round. Finally, in Section 3.3, we show that without computational assumptions, no four-round MPC can exist if broadcast is available only in the first, the third, and the last round.

3.1 Impossibility of $\texttt{p2p}^{(R-1)}\texttt{-bc}$ VSS

We show the impossibility of a *computational* VSS (without setup/PKI) with a R-round sharing phase (for any R) that uses broadcast only in the last round, when $n \leq 3t$.

Theorem 1. *There does not exist a computational VSS with a R-round sharing phase (with $R \geq 3$) that uses broadcast only in the last round, when $n \leq 3t$.*

Proof. Assume for contradiction that there exists a 3-party VSS scheme with a R-round sharing phase that uses broadcast only in the last round. We denote the parties running the protocol as P_1, P_2 and P_3, where P_1 acts as the dealer with input $b \in \{0, 1\}$. We assume that the last round involves only broadcast communication. This is without loss of generality, since any private communication in the last round can be emulated by exchanging pairwise random masks in the earlier rounds and broadcasting the masked message.

We start by providing a high-level description of the three relevant scenarios that the adversary we design will employ during the sharing phase. These scenarios are highlighted below (and sketched in Figure 1). A formal description of the scenarios is provided hereafter, along with more rigorous arguments.

Scenario 1 (P_1 corrupt, interacting with $P_2 := P_{2a}$ and $P_3 := P_{3a}$):
During the point-to-point rounds, the corrupt dealer (P_1) does not send any messages to P_{2a}, and ignores all messages received from P_{2a}.
Additionally, P_1 runs fictitious P_{2b} and P_{3b} in its head. These imaginary parties are both honest, and communicate with one another honestly, but P_1 does not send (in its own head) any messages to P_{3b}, and ignores all messages received from P_{3b}. The messages P_1 should have sent to P_{2a}, it instead sends (in its own head) to P_{2b}. P_1 sends its final broadcast "honestly", as though it had truly conversed with P_{3a} and imaginary P_{2b}.
Scenario 2 ($P_2 := P_{2a}$ corrupt, interacting with P_1 and $P_3 := P_{3a}$):
During the point-to-point rounds, the corrupt P_{2a} runs fictitious P_{2b} and P_{3b} in its head. These imaginary parties are both honest, and communicate with one another honestly, but P_{2a} does not send (in its own head) any messages to either party. Instead, it routes the messages it receives from P_1 to P_{2b}, and responds to P_1 accordingly. This ensures that the real parties (P_{2a}, P_{3a}) and the fictional parties (P_{2b}, P_{3b}) are only connected through P_1. P_{2a} sends its final broadcast "honestly", as though it had truly only conversed with P_{3a}.
Scenario 3 ($P_3 := P_{3a}$ corrupt, interacting with P_1 and $P_2 := P_{2b}$):
During the point-to-point rounds, the corrupt P_{3a} runs fictitious P_{2a} and P_{3b} in its head. These imaginary parties are both honest; P_{3a} itself talks (in its head) to P_{2a}, and forwards messages it receives from P_{2b} to P_{3b}. However, neither imaginary party hears anything from P_1. We maintain the invariant that (P_{2a}, P_{3a}) and (P_{2b}, P_{3b}) are only connected through P_1.
P_{3a} sends its final broadcast "honestly", as though it had truly only conversed with P_1 and imaginary P_{2a}.

Now, we consider what happens during reconstruction, and observe how this leads to a violation of the privacy property of the VSS scheme that we have assumed to be secure at the beginning of the proof.

Scenario 2 Corrupt P_{2a} participates in reconstruction consistently with its pretense of having communicated only with P_{3a}. By correctness, since the dealer (P_1) is honest, reconstruction must yield msg.

Scenario 1.1 Corrupt P_1 participates in reconstruction consistently with its pretense of having communicated with P_{3a} and imaginary P_{2b}. All reconstruction messages are distributed identically to those in Scenario 2, so reconstruction must yield msg.
Scenario 1.2 Corrupt P_1 does not participate in reconstruction. By commitment (since a corrupt dealer should not be able to change the outcome of reconstruction, even by refusing to participate), msg must still be recovered.
Scenario 3 Corrupt P_{3a} now has access to all information that was used in Scenario 1.2 to recover msg (since P_{2a} is now imaginary); thus, corrupt P_{3a} can violate privacy.

We now formalize the arguments sketched above. We recall that we denote with $\mathtt{sh}_i^{(\rho)}$ the algorithm executed by P_i in round ρ of the sharing phase, where $i \in [3], \rho \in [R]$. The output of $\mathtt{sh}_i^{(\rho)}$ consists of the set of the $\mathtt{p2p}$ messages and broadcast messages that need to be sent in round ρ.
Scenario 1. Consider an adversary $\mathcal{A}$ who controls a corrupt dealer P_1. $\mathcal{A}$ only controls P_1 throughout the protocol, using some fixed value $b \in \{0,1\}$ as input. $\mathcal{A}$ samples the bit-string $r_1^\star = r_1||r_{2b}||r_{3b}$ uniformly at random and instructs P_1 to behave as follows.

Round 1 ($\mathtt{p2p}$):
- Compute $(\mathsf{msg}_{1\to 2b}^{(1)}, \mathsf{msg}_{1\to 3}^{(1)}) \leftarrow \mathtt{sh}_1^{(1)}(b, r_1)$. Send $\mathsf{msg}_{1\to 3}^{(1)}$ to P_3 and send nothing to P_2. Record $\mathsf{msg}_{1\to 2b}^{(1)}$.
- Compute $(\mathsf{msg}_{2b\to 1}^{(1)}, \mathsf{msg}_{2b\to 3b}^{(1)}) \leftarrow \mathtt{sh}_2^{(1)}(r_{2b})$ honestly. Record $\mathsf{msg}_{2b\to 1}^{(1)}$ and $\mathsf{msg}_{2b\to 3b}^{(1)}$.
- Compute $(\mathsf{msg}_{3b\to 1}^{(1)}, \mathsf{msg}_{3b\to 2b}^{(1)}) \leftarrow \mathtt{sh}_3^{(1)}(r_{3b})$ honestly. Discard $\mathsf{msg}_{3b\to 1}^{(1)}$ and record $\mathsf{msg}_{3b\to 2b}^{(1)}$.

Round ρ for $\rho \in \{2, \ldots, R-1\}$ ($\mathtt{p2p}$): Let $\mathsf{msg}_{2\to 1}^{(\rho-1)}$ and $\mathsf{msg}_{3\to 1}^{(\rho-1)}$ be the messages received in round $\rho-1$ from P_2 and P_3, respectively.
- Ignore $\mathsf{msg}_{2\to 1}^{(\rho-1)}$.
- Define the view $\mathsf{V}_1^{(\rho-1)} = (b, r_1, (\mathsf{msg}_{2b\to 1}^{(j)}, \mathsf{msg}_{3\to 1}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{1\to 2b}^{(\rho)}, \mathsf{msg}_{1\to 3}^{(\rho)}) \leftarrow \mathtt{sh}_1^{(\rho)}(\mathsf{V}_1^{(\rho-1)})$. Record $\mathsf{msg}_{1\to 2b}^{(\rho)}$ and send $\mathsf{msg}_{1\to 3}^{(\rho)}$ to P_3.
- Define the view $\mathsf{V}_{2b}^{(\rho-1)} = (r_{2b}, (\mathsf{msg}_{1\to 2b}^{(j)}, \mathsf{msg}_{3b\to 2b}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{2b\to 1}^{(\rho)}, \mathsf{msg}_{2b\to 3b}^{(\rho)}) \leftarrow \mathtt{sh}_1^{(\rho)}(\mathsf{V}_{2b}^{(\rho-1)})$. Record $\mathsf{msg}_{2b\to 1}^{(\rho)}$ and $\mathsf{msg}_{2b\to 3b}^{(\rho)}$.
- Define the view $\mathsf{V}_{3b}^{(\rho-1)} = (r_{3b}, (\perp, \mathsf{msg}_{2b\to 3b}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{3b\to 1}^{(\rho)}, \mathsf{msg}_{3b\to 2b}^{(\rho)}) \leftarrow \mathtt{sh}_1^{(\rho)}(\mathsf{V}_{3b}^{(\rho-1)})$. Discard $\mathsf{msg}_{3b\to 1}^{(\rho)}$ and record $\mathsf{msg}_{3b\to 2b}^{(\rho)}$.

Round R ($\mathtt{bc}$): Upon receiving $\mathsf{msg}_{2\to 1}^{(R-1)}, \mathsf{msg}_{3\to 1}^{(R-1)}$ do the following:
- Define the view $\mathsf{V}_1^{(R-1)} = (b, r_1, (\mathsf{msg}_{2b\to 1}^{(j)}, \mathsf{msg}_{3\to 1}^{(j)})_{j\in[R-1]})$.
- Compute $\mathtt{bc}_1^{(R)} \leftarrow \mathtt{sh}_1^{(R)}(\mathsf{V}^{(R-1)})$. Broadcast $\mathtt{bc}_1^{(R)}$.

Reconstruction for Scenario 1. P_1 acts honestly during the reconstruction phase.

Scenario 1.1. This is exactly the same as Scenario 1, with the difference that P_1 does not participate in the reconstruction phase.

Lemma 1. *Let b^1 and $b^{1.1}$ represent the output of P_3 at the end of the reconstruction phase of Scenario* 1 *and of Scenario* 1.1*, respectively. Then* $\Pr[b^1 = b^{1.1}] \geq 1 - negl(\lambda)$.

Proof. Assume for contradiction the above does not hold (i.e., $b^1 \neq b^{1.1}$ with non-negligible probability). We can design a P_1 that breaks the commitment phase. Indeed, P_1 can decide, after the sharing phase, to either participate in the reconstruction phase, hence forcing the honest P_3 to output b^1, or to not participate in the reconstruction phase, forcing P_3 to output $b^{1.1}$.

Scenario 2. Consider an adversary $\mathcal{A}$ who controls a corrupt P_2. $\mathcal{A}$ samples the bit-string $r_2^* = r_{2a}||r_{2b}||r_{3b}$ uniformly at random and instructs P_2 to behave as follows.

Round 1 (p2p):
- Compute $(\mathsf{msg}^{(1)}_{2a\to 1}, \mathsf{msg}^{(1)}_{2a\to 3a}) \leftarrow \mathtt{sh}_2^{(1)}(r_{2a})$ honestly. Discard $\mathsf{msg}^{(1)}_{2a\to 1}$ and send $\mathsf{msg}^{(1)}_{2a\to 3a}$ to P_3.
- Compute $(\mathsf{msg}^{(1)}_{2b\to 1}, \mathsf{msg}^{(1)}_{2b\to 3b}) \leftarrow \mathtt{sh}_2^{(1)}(r_{2b})$ honestly. Send $\mathsf{msg}^{(1)}_{2b\to 1}$ to P_1 and record $\mathsf{msg}^{(1)}_{2b\to 3b}$.
- Compute $(\mathsf{msg}^{(1)}_{3b\to 1}, \mathsf{msg}^{(1)}_{3b\to 2b}) \leftarrow \mathtt{sh}_3^{(1)}(r_{3b})$ honestly. Discard $\mathsf{msg}^{(1)}_{3b\to 1}$ and record $\mathsf{msg}^{(1)}_{3b\to 2b}$

Round ρ for $\rho \in \{2, \ldots, R-1\}$ (p2p): Let $\mathsf{msg}^{(\rho-1)}_{1\to 2}$ and $\mathsf{msg}^{(\rho-1)}_{3\to 2}$ be the messages received in round $\rho - 1$ from P_1 and P_3, respectively.
- Ignore $\mathsf{msg}^{(\rho-1)}_{1\to 2}$.
- Define the view $\mathsf{V}^{(\rho-1)}_{2a} = (r_{2a}, (\perp, \mathsf{msg}^{(j)}_{3\to 2})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}^{(\rho)}_{2a\to 1}, \mathsf{msg}^{(\rho)}_{2a\to 3a}) \leftarrow \mathtt{sh}_2^{(\rho)}(\mathsf{V}^{(\rho-1)}_{2a})$. Discard $\mathsf{msg}^{(\rho)}_{2a\to 1}$ and send $\mathsf{msg}^{(\rho)}_{2a\to 3a}$ to P_3.
- Define the view $\mathsf{V}^{(\rho-1)}_{2b} = (r_{2b}, (\mathsf{msg}^{(j)}_{1\to 2b}, \mathsf{msg}^{(j)}_{3b\to 2b})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}^{(\rho)}_{2b\to 1}, \mathsf{msg}^{(\rho)}_{2b\to 3b}) \leftarrow \mathtt{sh}_2^{(\rho)}(\mathsf{V}^{(\rho-1)}_{2b})$. Send $\mathsf{msg}^{(\rho)}_{2b\to 1}$ to P_1 and record $\mathsf{msg}^{(\rho)}_{2b\to 3b}$.
- Define the view $\mathsf{V}^{(\rho-1)}_{3b} = (r_{3b}, (\perp, \mathsf{msg}^{(j)}_{2b\to 3b})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}^{(\rho)}_{3b\to 1}, \mathsf{msg}^{(\rho)}_{3b\to 2b}) \leftarrow \mathtt{sh}_1^{(\rho)}(\mathsf{V}^{(\rho-1)}_{3b})$. Discard $\mathsf{msg}^{(\rho)}_{3b\to 1}$ and record $\mathsf{msg}^{(\rho)}_{3b\to 2b}$.

Round R (bc): Upon receiving $\mathsf{msg}^{(R)}_{1\to 2}, \mathsf{msg}^{(R)}_{3\to 2}$ do the following:
- Set $\mathsf{V}^{(R-1)}_{2a} = (r_{2a}, (\perp, \mathsf{msg}^{(j)}_{3\to 2})_{j\in[R-1]})$.
- Compute $\mathtt{bc}^R_{2a} \leftarrow \mathtt{sh}_2^{(R)}(\mathsf{V}^{(R-1)}_{2a})$. Broadcast $\mathtt{bc}^{(R)}_{2a}$.

Reconstruction for Scenario 2. P_2 acts honestly during the reconstruction phase using the view of P_{2a}.

Lemma 2. *Let $b^{1.1}$ and b^2 represent the outputs of P_3 at the end of the reconstruction phase of Scenario* 1.1 *and of Scenario* 2*, respectively. Then* $\Pr[b^{1.1} = b^2] \geq 1 - negl(\lambda)$.

Proof. We first observe that the probability distribution of the view of P_3 in Scenario 1 is identical to that of P_3 in Scenario 2. This observation, paired with Lemma 1, concludes the proof.

Scenario 3. Consider an adversary $\mathcal{A}$ who controls P_3. $\mathcal{A}$ samples the bit string $r_3^* = r_{2a}||r_{3a}||r_{3b}$ uniformly at random and instructs P_3 to behave as follows.

Round 1 (p2p):

- Compute the first-round messages $(\mathsf{msg}_{2a\to 1}^{(1)}, \mathsf{msg}_{2a\to 3a}^{(1)}) \leftarrow \mathtt{sh}_2^{(1)}(r_{2a})$ honestly. Discard $\mathsf{msg}_{2a\to 1}^{(1)}$ and record $\mathsf{msg}_{2a\to 3a}^{(1)}$.
- Compute the first-round messages $(\mathsf{msg}_{3a\to 1}^{(1)}, \mathsf{msg}_{3a\to 2a}^{(1)}) \leftarrow \mathtt{sh}_3^{(1)}(r_{3a})$ honestly. Send $\mathsf{msg}_{3a\to 1}^{(1)}$ to P_1 and record $\mathsf{msg}_{3a\to 2a}^{(1)}$.
- Compute the first-round messages $(\mathsf{msg}_{3b\to 1}^{(1)}, \mathsf{msg}_{3b\to 2b}^{(1)}) \leftarrow \mathtt{sh}_3^{(1)}(r_{3b})$ honestly. Discard $\mathsf{msg}_{3b\to 1}^{(1)}$ and send $\mathsf{msg}_{3b\to 2b}^{(1)}$ to P_2.

Round ρ for $\rho \in \{2, \ldots, R-1\}$ (p2p): Let $\mathsf{msg}_{1\to 3}^{(\rho-1)}$ and $\mathsf{msg}_{2\to 3}^{(\rho-1)}$ be the messages received from P_1 and P_2 respectively.

- Define the view $\mathsf{V}_{2a}^{(\rho-1)} = (r_{2a}, (\perp, \mathsf{msg}_{3a\to 2a}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{2a\to 1}^{(\rho)}, \mathsf{msg}_{2a\to 3a}^{(\rho)}) \leftarrow \mathtt{sh}_2^{(\rho)}(\mathsf{V}_{2a}^{(\rho-1)})$. Discard $\mathsf{msg}_{2a\to 1}^{(\rho)}$ and record $\mathsf{msg}_{2a\to 3a}^{(\rho)}$.
- Define the view $\mathsf{V}_{3a}^{(\rho-1)} = (r_{3a}, (\mathsf{msg}_{1\to 3}^{(j)}, \mathsf{msg}_{2a\to 3a}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{3a\to 1}^{(\rho)}, \mathsf{msg}_{3a\to 2a}^{(\rho)}) \leftarrow \mathtt{sh}_3^{(\rho)}(\mathsf{V}_{3a}^{(\rho-1)})$. Send $\mathsf{msg}_{3a\to 1}^{(\rho)}$ to P_1 and record $\mathsf{msg}_{3a\to 2a}^{(\rho)}$.
- Define the view $\mathsf{V}_{3b}^{(\rho-1)} = (r_{3b}, (\perp, \mathsf{msg}_{2\to 3}^{(j)})_{j\in[\rho-1]})$ and compute $(\mathsf{msg}_{3b\to 1}^{(\rho)}, \mathsf{msg}_{3b\to 2b)}^{(\rho)} \leftarrow \mathtt{sh}_3^{(\rho)}(\mathsf{V}_{3b}^{(\rho-1)})$. Discard $\mathsf{msg}_{3b\to 1}^{(\rho)}$ and send $\mathsf{msg}_{3b\to 2b}^{(\rho)}$ to P_2.

Round R (bc): P_3 stops and does not send any messages during the last round.

In Lemma 2, we have proven that P_3 reconstructs to the same value in Scenario 1.1 and Scenario 2. In particular, this means that P_3 can reconstruct the correct output without the contribution of P_1 in the reconstruction phase. More formally, from Lemma 2 we have that, in Scenario 2, the honest P_3 could compute the output using only the following: $(r_3, (\mathsf{msg}_{1\to 3}^{(j)}, \mathsf{msg}_{2a\to 3a}^{(j)})_{j\in[R-1]}, \mathtt{bc}_1^{(R)}, \mathtt{bc}_2^{(R)}, \mathtt{rec}_2)$, where $\mathtt{rec}_2$ represents all the reconstruction phase messages generated by a P_2 that did not hear from P_1 during either phase.

We now observe that the distribution of the view of P_{2a} in Scenario 2 is identical to that of P_{2a} in Scenario 3 (where P_{2a} is emulated by $\mathcal{A}$ along with P_{3a}). Hence, P_3 in Scenario 3 can reconstruct the input of P_1 with the contribution of neither P_1 nor P_2 during the reconstruction phase. In particular, the adversary of Scenario 3 only needs the view of P_{3a} and P_{2a} to reconstruct the input of P_1 (that we recall are both available to $\mathcal{A}$ of Scenario 3). This implies that the VSS scheme cannot be private, which yields a contradiction.

3.2 Impossibility of bc-p2p$^{(R-2)}$-bc VSS

We show the impossibility of an *information-theoretic* VSS (without setup/PKI) with an R-round sharing phase (with $R \geq 3$) that uses broadcast only in the first and last rounds when $n \leq 3t$.

Theorem 2. *There does not exist a statistical VSS with an R-round sharing phase (with $R \geq 3$) that uses broadcast only in the first and last round when $n \leq 3t$.*

Proof. The proof is roughly the same as that of Theorem 1. As before, we assume for contradiction the existence of a secure 3-party VSS scheme that uses broadcast only in the first and last rounds, and show an adversary that breaks such a scheme. We consider three scenarios that are almost identical to the scenarios considered in the proof Theorem 1, with minor differences that we highlight below.

Scenario 1 (P_1 corrupt, interacting with $P_2 := P_{2a}$ and $P_3 := P_{3a}$): This is exactly the same as Scenario 1 considered in the proof of Theorem 1 with the following difference. The rushing adversary $\mathcal{A}$, instead of sampling a uniformly random $r_1^\star$, waits to receive the first-round broadcast messages $(\mathtt{bc}_2^{(1)}, \mathtt{bc}_3^{(1)})$. $\mathcal{A}$ then samples r_{2b} and r_{3b} such that the following hold:

- $(\mathtt{bc}_3^{(1)}, \mathsf{msg}_{3b\to 1}^{(1)}, \mathsf{msg}_{3b\to 2b}^{(1)}) = \mathtt{sh}_3^{(1)}(r_{3b})$ and
- $(\mathtt{bc}_2^{(1)}, \mathsf{msg}_{2b\to 1}^{(1)}, \mathsf{msg}_{2b\to 3b}^{(1)}) = \mathtt{sh}_2^{(1)}(r_{2b})$.

$\mathcal{A}$ then samples a uniformly random r_1, sets $r_1^\star := r_1||r_{2b}||r_{3b}$, and follows exactly the same steps described in Scenario 1, but using $r_1^\star$. In a nutshell, we make sure that the emulated P_{2b} and P_{3b} compute the point-to-point messages after having generated broadcast messages consistent with what the real P_2 and P_3 have generated. The point-to-point messages generated by P_{2b} and P_{3b} may (and most likely will) differ from the messages generated by the real P_2 and P_3.

We note that it is always possible to construct such $r_1^\star$. This process may not be efficient, but our aim is to find an attacker $\mathcal{A}$ for the VSS scheme that we are assuming to be (statistically) secure against unbounded adversaries.

Scenario 2 ($P_2 := P_{2a}$ corrupt, interacting with P_1 and $P_3 := P_{3a}$): This is exactly the same as Scenario 2 considered in the proof of Theorem 1 with the following difference. The rushing adversary $\mathcal{A}$, instead of sampling a uniformly random $r_2^\star$, waits to receive the first-round broadcast messages $(\mathtt{bc}_1^{(1)}, \mathtt{bc}_3^{(1)})$. $\mathcal{A}$ samples r_{2a} uniformly at random, computes $(\mathtt{bc}_2^{(1)}, \mathsf{msg}_{2a\to 1}^{(1)}, \mathsf{msg}_{2a\to 3a}^{(1)}) \leftarrow \mathtt{sh}_2^{(1)}(r_{2a})$, and samples r_{2b} and r_{3b} such that the following hold:

- $(\mathtt{bc}_2^{(1)}, \mathsf{msg}_{2b\to 1}^{(1)}, \mathsf{msg}_{2b\to 3b}^{(1)}) = \mathtt{sh}_2^{(1)}(r_{2b})$ and
- $(\mathtt{bc}_3^{(1)}, \mathsf{msg}_{3b\to 1}^{(1)}, \mathsf{msg}_{3b\to 2b}^{(1)}) = \mathtt{sh}_3^{(1)}(r_{3b})$.

$\mathcal{A}$ then sets $r_2^\star := r_{2a}||r_{2b}||r_{3b}$, and follows exactly the same steps described in Scenario 2, but using $r_2^\star$.

Scenario 3 ($P_3 := P_{3a}$ corrupt, interacting with P_1 and $P_2 := P_{2b}$): This is exactly the same as Scenario 3 considered in the proof of Theorem 1 with the following difference. The rushing adversary $\mathcal{A}$, instead of sampling a uniformly random $r_3^\star$, waits to receive the first-round broadcast messages $(\mathtt{bc}_1^{(1)}, \mathtt{bc}_2^{(1)})$. $\mathcal{A}$ samples r_{3a} uniformly at random, computes $(\mathtt{bc}_3^{(1)}, \mathtt{msg}_{3a\to 1}^{(1)}, \mathtt{msg}_{3a\to 2a}^{(1)}) \leftarrow \mathtt{sh}_3^{(1)}(r_{3a})$, and samples r_{2a}, r_{3b} such that

- $(\mathtt{bc}_2^{(1)}, \mathtt{msg}_{2a\to 1}^{(1)}, \mathtt{msg}_{2a\to 3a}^{(1)}) = \mathtt{sh}_2^{(1)}(r_{2a})$ and
- $(\mathtt{bc}_3^{(1)}, \mathtt{msg}_{3b\to 1}^{(1)}, \mathtt{msg}_{3b\to 2b}^{(1)}) = \mathtt{sh}_3^{(1)}(r_{3b})$.

$\mathcal{A}$ then sets $r_3^\star := r_{2a}||r_{3a}||r_{3b}$, and follows exactly the same steps described in Scenario 3, but using $r_3^\star$ as we have just defined.

The rest of the proof follows by using exactly the same arguments of Theorem 1, which leads to the adversary of Scenario 3 breaking the privacy of the VSS scheme.

3.3 Impossibility of bc-p2p$^{(R-3)}$-bc-bc MPC

Consider a three-party functionality $f_{OT}(b \in \{0,1\}, \perp, (x_0, x_1)) \to (x_b, \perp, \perp)$ with $b, x_0, x_1 \in \{0,1\}$

Theorem 3. *There does not exist a statistically secure straight-line-simulatable R-round MPC (with $R \geq 4$) protocol that uses broadcast only in the first, the second to last (penultimate), and the last round, that securely evaluates f_{OT} when $n \leq 3t$.*

Proof. Our proof shares a common structure with that of Theorem 1 and Theorem 2. We provide a high-level description of the proof for the pattern bc-p2p-bc-bc, and the full version [CDR+25] contains the formal argument for the more general pattern bc-p2p$^{(R-3)}$-bc-bc for any $R \geq 4$.

For the sake of contradiction, assume that there exists a protocol Π_{OT} that realizes f_{OT} with the pattern bc-p2p-bc-bc. Let $\mathcal{S}$ be the simulator for Π_{OT}. We note that if P_1 is corrupt, $\mathcal{S}$ should be able to extract b after the third round. This is because the simulator will need to send b to the ideal functionality and receive x_b in time to manipulate the final-round messages so as to induce x_b as P_1's output (we will provide a formal argument about this fact). Consider now the following sequence of scenarios:

Scenario 1 (P_1 corrupt, interacting with P_{2a} and P_{3a}): During the first two rounds, the corrupt P_1 does not send P_{2a} any point-to-point messages, and ignores all point-to-point messages received from P_{2a}. Additionally, P_1 runs fictitious P_{2b} and P_{3b} in its head (exactly as in Scenario 1 from the proof of Theorem 2).
P_1 sends its third and fourth-round broadcasts "honestly", as though it had truly conversed with P_{3a} and imaginary P_{2b} over point-to-point channels.

Scenario 2 (P_{2a} corrupt, interacting with P_1 and P_{3a}): During the first two rounds, the corrupt P_{2a} does not send P_1 any point-to-point messages. It runs fictitious P_{2b} and P_{3b} in its head (exactly as in Scenario 2 from the proof of Theorem 2). P_{2a} samples randomness on behalf of P_{2b} that is consistent with its own first-round broadcast, and randomness *and input* on behalf of P_{3b} that is consistent with the first-round broadcast of P_{3a}. It computes their first and second-round point-to-point messages based on that. Fictitious P_{2b} and P_{3b} are both honest, and communicate with one another honestly, but P_{2a} does not send (in its own head) any messages to either party. Instead, it routes the messages it receives from P_1 to P_{2b}, and responds to P_1 accordingly. (This ensures that the real parties (P_{2a}, P_{3a}) and the fictional parties (P_{2b}, P_{3b}) are only connected through P_1.)
P_{2a} sends its third and fourth-round broadcasts "honestly", as though it had truly only conversed with P_{3a} over point-to-point channels.
We now make the following observations.

1. By correctness, in Scenario 2 P_1 must receive the correct output x_b.
2. In Scenario 1, because corrupt P_1's view subsumes that of honest P_1 in Scenario 2, corrupt P_1 must learn x_b. Moreover, the simulator will see $\mathsf{bc}_1^{(1)}$, $\mathsf{msg}_{1\to 3}^{(1)}$, $\mathsf{msg}_{1\to 3}^{(2)}$ and $\mathsf{bc}_1^{(3)}$ (as well as its own state) before the last round of broadcast, by which point it must be able to extract b.

Scenario 3 (P_{3a} corrupt, interacting with P_1 and P_{2b}): During the first two rounds, the corrupt P_{3a} does not send P_{2b} any point-to-point messages. It runs fictitious P_{2a} and P_{3b} in its head (exactly as in Scenario 3 from the proof of Theorem 2). P_{3a} samples randomness and input on behalf of P_{3b} that is consistent with its own first-round broadcast, and randomness on behalf of P_{2a} that is consistent with the first-round broadcast of P_{2b}. It computes their first and second-round point-to-point messages based on that. These imaginary parties are both honest; P_{3a} talks (in its head) to P_{2a}, and forwards messages it receives from P_{2b} to P_{3b}. However, neither imaginary party hears anything from P_1, and they do not talk to one another.
P_3 is rushing in the third round and waits to receive the other parties' third round broadcasts, whereupon it stops. *Note that in this scenario, corrupt P_{3a} sees all of $\mathsf{bc}_1^{(1)}$, $\mathsf{msg}_{1\to 3}^{(1)}$, $\mathsf{msg}_{1\to 3}^{(2)}$ and $\mathsf{bc}_1^{(3)}$.*
The final adversary (P_{3a} corrupt, interacting with P_1 and P_{2b}): In this scenario, corrupt P_{3a} imagines P_{3b} as before (in Scenario 3). However, instead of running itself and P_{2a}, *it runs the simulator $\mathcal{S}$ to generate the messages P_{3a} and P_{2a} would have sent in Scenario 2.* Now, P_{3a} has access to the simulator's entire state and is able to extract b, violating the security of Π_{OT}.

The formal argument is provided in the full version of the paper [CDR+25].

4 No Broadcast in the Third Round

In this section, we prove that statistically-secure three-round VSS (Section 4.1) and statistically-secure four-round MPC (Section 4.2) both require broadcast in the third round.

4.1 Impossibility of bc-bc-p2p VSS

We show the impossibility of statistical VSS with a three-round sharing phase that does not use broadcast in the last round in the setting where $2t < n \leq 2.5t$. More specifically, we present the argument for $n = 5$ and $t = 2$, and then show that this suffices to rule out VSS for the above range of thresholds.

We begin with a high-level description of the proof. For contradiction, suppose $\Pi_{\mathtt{VSS}}$ is a secure VSS protocol with a 3-round sharing phase that uses only point-to-point channels in the last round. We first consider Scenario 1 where the dealer P_1 is corrupt and acts honestly with input bit σ except for the following misbehavior: It ignores the private communication from P_3 and P_4. Instead, it (1) creates a fake view by sampling randomness $\overline{r_3}$ and $\overline{r_4}$ on behalf of P_3 and P_4 respectively such that it is consistent with their first round broadcast message, (2) pretends that P_3 and P_4 sent its first round point-to-point message using $\overline{r_3}$ and $\overline{r_4}$ respectively. Now, from the perspective of P_2 and P_5, this scenario is indistinguishable from another scenario where P_3 and P_4 were corrupt and sent a point-to-point message to P_1 using a different randomness than the one used for communication with other parties. In the latter case since the scenario should lead to reconstruction of σ as the dealer is honest, so should the former case.

Next, we claim that the combined view of the honest parties at the end of Round 2 in Scenario 1 must be independent of σ. Intuitively, this is because (a) P_3 and P_4 never heard any private information from P_1, so whatever information is available to them about σ is only via P_1's broadcast which is also available to P_2 and P_5 and (b) P_1's broadcast messages in the first and second round and its first round point-to-point messages towards P_2 and P_5 should still protect P_1's input σ – as these are messages an honest P_1 would send when the randomness used by, say, honest P_3 and P_4, was in fact $\overline{r_3}$ and $\overline{r_4}$. Otherwise privacy against potentially (semi-honest) corrupt P_2 and P_5 would be violated.

We now fork Scenario 1 into various possibilities based on whether the adversary communicates to P_2 and P_5 over the third round point-to-point channel or not. We show via a sequence of hybrid scenarios exploiting the commitment property of the VSS that even if the adversary stays silent in the third round, the secret reconstructed should still be σ. This leads us to our final contradiction as the claim above shows that the views of the parties at the end of the sharing phase does not suffice to reconstruct σ.

We present the formal theorem and the detailed proof below.

Theorem 4. *There does not exist a* 5*-party statistical VSS with a* 3*-round sharing phase that does not use broadcast in the last round and achieves security against* $t = 2$ *corruptions.*

Proof. Towards a contradiction, assume a 5-party statistical VSS protocol $\Pi_{\mathtt{VSS}}$ with a 3-round sharing phase that uses only point-to-point channels in the last round. More specifically, we assume that $\Pi_{\mathtt{VSS}}$ satisfies ϵ-correctness and ϵ-privacy, where ϵ is negligible. For any communication beyond Round 1 that is allowed to use both broadcast and point-to-point channels (such as Round 2 and

the reconstruction phase), we assume that this communication occur only over broadcast channels. Note that this is without loss of generality, as the parties can exchange pairwise random masks over point-to-point channels in the first round, and broadcast the masked versions of any potential private communication in the subsequent rounds instead [GIKR01].

Before describing the scenarios, we introduce the following notation: Let $\mathtt{sh}_i^{(\rho)}$ denote the algorithm executed by P_i in Round ρ of the sharing phase, where $i \in [5], \rho \in [3]$. As part of adversary strategies, we refer to 'conditional sampling' to mean that the adversary repeatedly samples randomness from a uniform distribution until the specific condition is satisfied. We now describe the scenarios.

Scenario 1: Consider an adversary $\mathcal{A}_\sigma$ (where $\sigma \in \{0,1\}$) who controls a corrupt dealer P_1. Below, we describe the instructions of $\mathcal{A}_\sigma$ to P_1.

Round 1 ($\mathtt{bc}+\mathtt{p2p}$): Sample randomness r_1, and use it to compute the first-round messages $(\mathtt{bc}_1^{(1)}, \{\mathsf{msg}_{1\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_1^{(1)}(\sigma; r_1)$ honestly for secret σ. Broadcast $\mathtt{bc}_1^{(1)}$, send $\mathsf{msg}_{1\to 2}^{(1)}$ to P_2 and $\mathsf{msg}_{1\to 5}^{(1)}$ P_5. Note that all of the messages are sent, *except for* $\mathsf{msg}_{1\to 3}^{(1)}$ *and* $\mathsf{msg}_{1\to 4}^{(1)}$.

Local Computation: Ignore the point-to-point message $\mathsf{msg}_{3\to 1}^{(1)}$ received from P_3 and $\mathsf{msg}_{4\to 1}^{(1)}$ received from P_4. Let $\mathtt{bc}_j^{(1)}$ denote the first round broadcast message of party P_j ($j \in [5]$). Conditionally sample randomness $\overline{r_3}$ on behalf of P_3 and $\overline{r_4}$ on behalf of P_4, such that is consistent with the broadcast messages $\mathtt{bc}_3^{(1)}$ and $\mathtt{bc}_4^{(1)}$ respectively. More formally, sample $\overline{r_3}$ conditioned on $\overline{\mathtt{bc}}_3^{(1)} = \mathtt{bc}_3^{(1)}$, where $(\overline{\mathtt{bc}}_3^{(1)}, \{\overline{\mathsf{msg}}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_3^{(1)}(\overline{r_3})$. Similarly, sample $\overline{r_4}$ conditioned on $\overline{\mathtt{bc}}_4^{(1)} = \mathtt{bc}_4^{(1)}$, where $(\overline{\mathtt{bc}}_4^{(1)}, \{\overline{\mathsf{msg}}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_4^{(1)}(\overline{r_4})$. Compute the 'fake' view as $\mathsf{V}_1 = (\sigma, r_1, \{\mathtt{bc}_j^{(1)}\}_{j\in[5]}, \{\mathsf{msg}_{j\to 1}^{(1)}\}_{j\in\{1,5\}} \cup \{\overline{\mathsf{msg}}_{3\to 1}^{(1)}, \overline{\mathsf{msg}}_{4\to 1}^{(1)}\})$. This is the honest view of P_1 except that the point-to-point message $\mathsf{msg}_{3\to 1}^{(1)}$ and $\mathsf{msg}_{4\to 1}^{(1)}$ has been replaced with $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ and $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ respectively.

Round 2 ($\mathtt{bc}$): Compute the second round message honestly on behalf of P_1 based on the fake view V_1 as $\mathtt{bc}_1^{(2)} \leftarrow \mathtt{sh}_1^{(2)}(\mathsf{V}_1)$. This is well-defined, since V_1 is a plausible view of an honest P_1 who received $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ and $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ from P_3 and P_4 respectively. Let $\{\mathtt{bc}_j^{(2)}\}_{j\in[5]}$ denote the set of broadcasts sent in this round.

Round 3 ($\mathtt{p2p}$): Let $\{\mathsf{msg}_{1\to j}^{(3)}\}_{j\in[5]} \leftarrow \mathtt{sh}_1^{(3)}(\mathsf{V}_1, \{\mathtt{bc}_j^{(2)}\}_{j\in[5]})$ denote the third-round point-to-point messages honestly computed by P_1 according to the fake view and the public broadcasts. Send $\mathsf{msg}_{1\to 2}^{(3)}$ to P_2 and $\mathsf{msg}_{1\to 5}^{(3)}$ to P_5. No messages are sent to P_3 and P_4.

Reconstruction (bc): Replace any point-to-point messages $\mathsf{msg}_{3\to 1}^{(3)}$ and $\mathsf{msg}_{4\to 1}^{(3)}$ received from P_3 and P_4 with $\perp$; that is, pretend not to have received communication from them in Round 3. Continue to play according to the fake view and the public broadcasts.

Scenario 2: Consider a scenario where P_1 is honest, with input σ. The adversary corrupts P_3 and P_4 and instructs them to do the following:

Local Computation: Sample random strings r_3 and r_4 on behalf of P_3 and P_4 respectively. Compute first round messages $(\mathsf{bc}_3^{(1)}, \{\mathsf{msg}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_3^{(1)}(r_3)$ and $(\mathsf{bc}_4^{(1)}, \{\mathsf{msg}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_4^{(1)}(r_4)$.
Conditionally sample another set of random strings $\overline{r_3}$ and $\overline{r_4}$ such that it is consistent with $\mathsf{bc}_3^{(1)}$ and $\mathsf{bc}_4^{(1)}$ respectively. More formally, suppose $(\overline{\mathsf{bc}}_3^{(1)}, \{\overline{\mathsf{msg}}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_3^{(1)}(\overline{r_3})$ such that $\overline{\mathsf{bc}}_3^{(1)} = \mathsf{bc}_3^{(1)}$ and $(\overline{\mathsf{bc}}_4^{(1)}, \{\overline{\mathsf{msg}}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{sh}_4^{(1)}(\overline{r_4})$ such that $\overline{\mathsf{bc}}_4^{(1)} = \mathsf{bc}_4^{(1)}$.

Round 1 (bc + p2p): Broadcast $\mathsf{bc}_3^{(1)}$, send $\mathsf{msg}_{3\to 2}^{(1)}$ to P_2, send $\mathsf{msg}_{3\to 5}^{(1)}$ to P_5, and send $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ to P_1. Similarly, on behalf of P_4, broadcast $\mathsf{bc}_4^{(1)}$, send $\mathsf{msg}_{4\to 2}^{(1)}$ to P_2, send $\mathsf{msg}_{4\to 5}^{(1)}$ to P_5, and send $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ to P_1.
After receiving the first round messages, prepare a fake view V_3 comprising of the randomness r_3 and all received broadcast and point-to-point messages, except that $\mathsf{msg}_{1\to 3}^{(1)}$ received from P_1 in Round 1 is replaced with $\perp$. Similarly prepare a fake view V_4 comprising of the randomness r_4 and all received broadcast and point-to-point messages, except that $\mathsf{msg}_{1\to 4}^{(1)}$ received from P_1 in Round 1 is replaced with $\perp$. In other words, P_3 and P_4 pretend that they did not receive any point-to-point message from P_1 in the first round.

Round 2 (bc): Behave honestly according to the fake view V_3 on behalf of P_3 and V_4 on behalf of P_4.

Round 3 (p2p): Continue to behave honestly according to the fake view and the public broadcasts, except do not send a point-to-point message to P_1.

Reconstruction (bc). Both P_3 and P_4 replace the point-to-point messages $\mathsf{msg}_{1\to 3}^{(3)}$ and $\mathsf{msg}_{1\to 4}^{(3)}$ received from P_1 with $\perp$; i.e., they pretend to have not received communication from P_1 in Round 3. Compute the output according to the fake views and the public broadcasts.

We now make the following claims:

Lemma 3. *The secret reconstructed in Scenario 1 is σ with probability at least $1 - \epsilon$.*

Proof. First, we observe that since the dealer P_1 is honest in Scenario 2, it follows from correctness that the secret reconstructed in Scenario 2 is σ except with probability ϵ. Next, Since the views of honest P_2 and P_5 are identically distributed in Scenarios 1 and 2, their output must be the same in both scenarios i.e. it must be σ with probability at least $1 - \epsilon$.

Lemma 4. *Let $\mathsf{V}^{\sigma}_{\mathtt{sh}}$ denote the random variable corresponding to the combined view of $\{P_2, P_3, P_4, P_5\}$ at the end of Round 2 of the sharing phase of Scenario 1 (with respect to σ). Then, $\mathsf{V}^{0}_{\mathtt{sh}} \approx^{\epsilon} \mathsf{V}^{1}_{\mathtt{sh}}$ must hold.*

Proof. We consider another scenario, say Scenario 3 below.

Scenario 3: Suppose the dealer P_1 is honest and the adversary corrupts P_2 and P_5. Suppose the honest dealer has input σ and let the randomness used by honest P_3 and P_4 be denoted by $\widetilde{r_3}$ and $\widetilde{r_4}$ respectively (these are unknown to the adversary). P_2 and P_5 behave honestly, except that at the end of Round 2 they do the following experiment in their head to emulate Scenario 1.

- Conditionally sample r_3^* such that it is consistent with the first round point-to-point messages $\mathsf{msg}^{(1)}_{3\to 2}$ and $\mathsf{msg}^{(1)}_{3\to 5}$ and first round broadcast message $\mathtt{bc}^{(1)}_3$ received from P_3. More formally, r_3^* is such that $(\mathtt{bc}^{(1)}_3, \mathsf{msg}^{*(1)}_{3\to 1}, \mathsf{msg}^{(1)}_{3\to 2}, \mathsf{msg}^{*(1)}_{3\to 4}, \mathsf{msg}^{(1)}_{3\to 5}) \leftarrow \mathtt{sh}^{(1)}_3(r_3^*)$.
- Similarly, conditionally sample r_4^* such that it is consistent with the first round point-to-point messages $\mathsf{msg}^{(1)}_{4\to 2}$ and $\mathsf{msg}^{(1)}_{4\to 5}$ and first round broadcast message $\mathtt{bc}^{(1)}_4$ received from P_4. More formally, r_4^* is such that $(\mathtt{bc}^{(1)}_4, \mathsf{msg}^{*(1)}_{4\to 1}, \mathsf{msg}^{(1)}_{4\to 2}, \mathsf{msg}^{*(1)}_{4\to 3}, \mathsf{msg}^{(1)}_{4\to 5}) \leftarrow \mathtt{sh}^{(1)}_4(r_4^*)$.
- Compute $\mathtt{bc}^{*(2)}_3 \leftarrow \mathtt{sh}^{(2)}_3(r_3^*, \{\mathtt{bc}^{(1)}_j\}_{j\in[5]}, \perp, \mathsf{msg}^{(1)}_{2\to 3}, \mathsf{msg}^{*(1)}_{4\to 3}, \mathsf{msg}^{(1)}_{5\to 3})$ i.e. the second round broadcast message an honest P_3 with randomness r_3^* would compute, had it not received any private first-round message from P_1. Note that $\mathsf{msg}^{(1)}_{2\to 3}$ and $\mathsf{msg}^{(1)}_{5\to 3}$ are known to the adversary corrupting P_2 and P_5. Further, $\mathsf{msg}^{*(1)}_{4\to 3}$ was sampled by the adversary itself in the above step.
- Similarly, compute $\mathtt{bc}^{*(2)}_4 \leftarrow \mathtt{sh}^{(2)}_4(r_4^*, \{\mathtt{bc}^{(1)}_j\}_{j\in[5]}, \perp, \mathsf{msg}^{(1)}_{2\to 4}, \mathsf{msg}^{*(1)}_{3\to 4}, \mathsf{msg}^{(1)}_{5\to 4})$ i.e. the second round broadcast message an honest P_4 with randomness r_3^* would compute had it not received any private first-round message from P_1. Note that $\mathsf{msg}^{(1)}_{2\to 4}$ and $\mathsf{msg}^{(1)}_{5\to 4}$ are known to the adversary corrupting P_2 and P_5. Further, $\mathsf{msg}^{*(1)}_{3\to 4}$ was sampled by the adversary itself in the above step.
- Consider the view $\mathsf{V}^* = \{r_2, r_3^*, r_4^*, r_5, \{\mathtt{bc}^{(1)}_j\}_{j\in[5]}, \{\mathtt{bc}^{(2)}_j\}_{j\in\{1,2,5\}}, \{\mathtt{bc}^{*(2)}_j\}_{j\in\{3,4\}}, \{\mathsf{msg}^{(1)}_{j\to 2}\}_{j\in[5]}, \{\mathsf{msg}^{(1)}_{j\to 5}\}_{j\in[5]}, (\{\mathsf{msg}^{(1)}_{j\to 3}\}_{j\in\{2,5\}}, \mathsf{msg}^{*(1)}_{4\to 3}), (\{\mathsf{msg}^{(1)}_{j\to 4}\}_{j\in\{2,5\}}, \mathsf{msg}^{*(1)}_{3\to 4})\}$. This comprises the view of P_2, P_3, P_4, P_5 where P_2 and P_5's views comprise of the messages obtained in Scenario 3, while P_3 and P_4's messages are computed using the randomness r_3^* and r_4^* instead.

We observe that V^* must be distributed identically to $\mathsf{V}^{\sigma}_{\mathtt{sh}}$ by noting that:

- $\widetilde{r_i}(i \in \{3,4\})$ (the actual randomness used by parties P_3 and P_4) in Scenario 3 is essentially analogous to $\overline{r_i}(i \in \{3,4\})$ (the randomness sampled by the adversary corrupting P_1 in Scenario 1). Note that in both cases, P_1's second round messages are computed with respect to messages received based on this randomness.
- $r_i^*(i \in \{3,4\})$ (the randomness sampled by the adversary in Scenario 3 on behalf of P_3 and P_4) in Scenario 3 is analogous to $r_i(i \in \{3,4\})$ (the actual randomness used by honest P_3 and P_4 in Scenario 1). Note that in both cases, this is the randomness with respect to which parties P_3 and P_4's messages are computed.

The main idea is that since in Scenario 1, P_1's messages were disentangled from the actual randomness used by $\{P_3, P_4\}$ (as P_1 ignores the point-to-point message received and computes its response imagining a different randomness), it becomes possible for P_2 and P_5 to essentially create a view in their head that is identically distributed to the view of $\{P_3, P_4\}$ in Scenario 1. This is possible just using the information they have, as shown above.

We note that since the dealer is honest in Scenario 3, the adversary corrupting P_2 and P_5 should not be able to create a view V^* that depends on the secret σ. Therefore, one can conclude from privacy of the VSS that V^* must be independent of the secret σ. Finally, since V^* is distributed identically to $\mathsf{V}^{\sigma}_{\mathtt{sh}}$ as shown above, $\mathsf{V}^{0}_{\mathtt{sh}} \approx^{\epsilon} \mathsf{V}^{1}_{\mathtt{sh}}$ must hold.

Lemma 5. *Consider Scenario 1' that is identical to Scenario 1, except that the adversary corrupting P_1 does not communicate in Round 3 and during reconstruction phase. Then, Scenario 1' must lead to reconstruction of σ with overwhelming probability.*

Proof. For this argument, consider the following sequence of hybrid scenarios and inferences.

Scenario 1A (P_1, P_5 corrupt): This is identical to Scenario 1, except that the adversary also corrupts P_5 and does the following: P_5 behaves honestly during the sharing phase and pretends during reconstruction that it did not receive point-to-point message from P_1 in the third round. P_1 stays silent during reconstruction.
Since the only difference between Scenario 1 and Scenario 1A is in the adversary's behaviour during reconstruction, it follows from the commitment property of VSS that this must not influence the reconstructed secret. Since Scenario 1 leads to reconstruction of σ (Lemma 3), we can infer that Scenario 1A must lead to reconstruction of σ as well (with overwhelming probability).

Scenario 1B (P_1 corrupt): This is identical to Scenario 1, except that in the third round P_1 drops its message to P_5 and stays silent during reconstruction.
We observe that Scenario 1B must also lead to reconstruction of σ (with overwhelming probability), as the views of honest P_2, P_3 and P_4 are identically

distributed to their respective views in Scenario 1A. Note that it is crucial here that the third round uses only point-to-point channels.

Scenario 1C (P_1, P_2 corrupt): This is identical to Scenario 1B except that the adversary also corrupt P_2 and does the following on behalf of P_2: Behave honestly on behalf of P_2 in the sharing phase but pretend during reconstruction that P_2 did not receive a point-to-point message from P_1 in the third round. The misbehaviour on behalf of P_1 is identical to Scenario 1B (i.e. drop the point-to-point message from P_1 to P_5 during the third round and stay silent during reconstruction.)
Since the only difference between Scenario 1C and Scenario 1B is in the adversary's behaviour during reconstruction, it follows from the commitment property of VSS that this must not influence the reconstructed secret. We can thus infer that Scenario 1C must lead to reconstruction of σ as well (with overwhelming probability).

Scenario 1' (P_1 corrupt): This is identical to Scenario 1, except that the adversary corrupting P_1 remains silent in the third round and reconstruction phase.
Lastly, Scenario 1' must also lead to reconstruction of σ (with overwhelming probability) because the views of honest P_3, P_4 and P_5 are identically distributed in Scenarios 1' and 1C.

We are now ready to present our final contradiction. Recall that the combined view of $\{P_2, P_3, P_4, P_5\}$ at the end of Round 2 of sharing phase in Scenario 1 is independent of σ (Lemma 4). Note that this is identically distributed to the combined view of the honest parties at the end of the (three rounds of) sharing phase in Scenario 1', as P_1 did not communicate after Round 2. Therefore, this cannot lead to reconstruction of σ, which contradicts our inference in Lemma 5. This completes the proof.

Theorem 5. *There does not exist a n-party statistical VSS with a 3-round sharing phase that does not use broadcast in the last round and achieves security against t corruptions where $2t < n \leq 2.5t$.*

Proof. We show the result via a standard player partitioning argument.

Now assume we have a VSS protocol π that works for n players where $n \leq 2.5t$, with a 3-round sharing phase that does not use broadcast in the last round. Assume for simplicity that $n = 2.5t$ (this is the hardest case). We transform this to a 5-player VSS contradicting Theorem 4 as follows: Split the n players into 5 disjoint subsets A_i for $i = 1, \ldots, 5$ such that A_1 contains the dealer, and such that each A_i has size $n/5$. The claimed 5-player VSS will execute π by having P_i emulate the players in A_i in his head. Since π is secure for t corruptions, it is secure if players in any pair of subsets are corrupted, and this clearly means we contradict Theorem 4. In general, if n is even smaller than $2.5t$, we can do something similar except we choose smaller sizes for the A_i's, so the sum of subset sizes is n.

4.2 Impossibility of bc-bc-p2p-bc MPC

In this section, we show that does not exist a four-round MPC with G.O.D. ϵ-security without broadcast in the third round when $n = 5$ and $t = 2$ which computes the functionality f described below:

$$f(\sigma, (m_0, m_1), (m'_0, m'_1), \perp, \perp) \rightarrow (m_\sigma, m'_\sigma, m_\sigma, m_\sigma, m'_\sigma)$$

At a high level, the proof begins with considering the same scenarios as in Theorem 4 with some modifications to extend to MPC. Intuitively, the arguments in Theorem 4 show that it is possible for P_1 to remain 'uncommitted' to its input σ until the end of the first two rounds. In the context of MPC, we show that if further this corrupt P_1 communicates to only one of the parties, say P_2 in Round 3 and remains silent in the last round (4th round) of the MPC protocol, the G.O.D. security guarantees that the honest parties must still obtain the output computed with respect to σ – otherwise, there exists an adversarial strategy that lets the adversary obtain two different evaluations of f, contradicting privacy. Finally, we show that simulating the above scenario where P_2 is additionally corrupt (but behaves honestly throughout) is not possible – as no straight-line simulator can extract σ based on the views of honest $\{P_3, P_4, P_5\}$. Intuitively, this is because (a) the combined views of $\{P_2, P_3, P_4, P_5\}$ at the end of Round 2 do not contain information about σ (analogous to Lemma 4 in the VSS case) and (b) P_3, P_4, P_5 never received any communication from P_1 after Round 2.

We state the formal theorem below.

Theorem 6. *There exists a function f such that no statistically secure 5-party 4-round MPC protocol can compute f with G.O.D. against $t = 2$ corruptions, where the third round uses only point-to-point channels.*

Proof. Towards a contradiction, assume a 5-party statistically-secure 4-round MPC protocol Π_f computing the function f (defined as above) that uses only point-to-point channels in the 3rd round and is secure against $t = 2$ corruptions. More specifically, we assume that Π_f satisfies ϵ-security, where ϵ is negligible. Similar to Theorem 4, we assume w.l.o.g. that the communication in Round 2 and Round 4 occurs only over broadcast channels (as masked versions of the private communication can be broadcast using random pads exchanged privately in the first round). We describe a sequence of scenarios below, which are identical to Theorem 4 for the first three rounds[8].

> **Scenario 1:** This scenario proceeds identically to Scenario 1 in Theorem 4 for the first three rounds, where the adversary $\mathcal{A}_\sigma$ corrupts P_1, with the additional modification that P_1 is silent in Round 4. We describe $\mathcal{A}_\sigma$'s misbehavior on behalf of P_1 below.

[8]Similar to Theorem 3 in the description below, the adversary makes use of the knowledge of the input of all the parties, corrupt and honest. Note that the definition of MPC does allow the adversary to know such inputs, as security must hold for any possible choice of inputs, even inputs known or decided adversarially.

Round 1 ($\mathtt{bc} + \mathtt{p2p}$): Sample randomness r_1, and use it to compute the first-round messages $(\mathtt{bc}_1^{(1)}, \{\mathsf{msg}_{1\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_1^{(1)}(\sigma; r_1)$ honestly for secret σ. Broadcast $\mathtt{bc}_1^{(1)}$ and send $\mathsf{msg}_{1\to j}^{(1)}$ to $j \in \{2, 5\}$. Note that all of the messages are sent, *except for* $\mathsf{msg}_{1\to 3}^{(1)}$ *and* $\mathsf{msg}_{1\to 4}^{(1)}$.

Local Computation: Ignore the point-to-point message $\mathsf{msg}_{3\to 1}^{(1)}$ received from P_3 and $\mathsf{msg}_{4\to 1}^{(1)}$ received from P_4. Let $\mathtt{bc}_j^{(1)}$ denote the first round broadcast message of party P_j ($j \in [5]$). Sample randomness $\overline{r_3}$ on behalf of P_3 and $\overline{r_4}$ on behalf of P_4, such that is consistent with the broadcast messages $\mathtt{bc}_3^{(1)}$ and $\mathtt{bc}_4^{(1)}$ respectively.
More formally, conditionally sample $\overline{r_3}$ such that $\overline{\mathtt{bc}}_3^{(1)} = \mathtt{bc}_3^{(1)}$, where $(\overline{\mathtt{bc}}_3^{(1)}, \{\overline{\mathsf{msg}}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_3^{(1)}((m_0', m_1'); \overline{r_3})$. Similarly, conditionally sample $\overline{r_4}$ such that $\overline{\mathtt{bc}}_4^{(1)} = \mathtt{bc}_4^{(1)}$, where $(\overline{\mathtt{bc}}_4^{(1)}, \{\overline{\mathsf{msg}}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_4^{(1)}(\overline{r_4})$.
Compute the 'fake' view as $\mathsf{V}_1 = (\sigma, r_1, \{\mathtt{bc}_j^{(1)}\}_{j\in[5]}, \{\mathsf{msg}_{j\to 1}^{(1)}\}_{j\in\{2,5\}} \cup \{\overline{\mathsf{msg}}_{3\to 1}^{(1)}, \overline{\mathsf{msg}}_{4\to 1}^{(1)}\})$. This is the honest view of P_1 except that the point-to-point messages $\mathsf{msg}_{3\to 1}^{(1)}$ and $\mathsf{msg}_{4\to 1}^{(1)}$ have been replaced with $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ and $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ respectively.

Round 2 ($\mathtt{bc}$): Compute the second round message honestly on behalf of P_1 based on the fake view V_1 as $\mathtt{bc}_1^{(2)} \leftarrow \mathtt{nxt\text{-}msg}_1^{(2)}(\mathsf{V}_1)$. This is well-defined, since V_1 is a plausible view of an honest P_1 who received $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ and $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ from P_3 and P_4 respectively. Let $\{\mathtt{bc}_j^{(2)}\}_{j\in[5]}$ denote the set of broadcasts sent in this round.

Round 3 ($\mathtt{p2p}$): Let $\{\mathsf{msg}_{1\to j}^{(3)}\}_{j\in[5]} \leftarrow \mathtt{nxt\text{-}msg}_1^{(3)}(\mathsf{V}_1, \{\mathtt{bc}_j^{(2)}\}_{j\in[5]})$ denote the third-round point-to-point messages honestly computed by P_1 according to the fake view and the public broadcasts. Send $\mathsf{msg}_{1\to 2}^{(3)}$ to P_2 and $\mathsf{msg}_{1\to 5}^{(3)}$ to P_5. No messages are sent to P_3 and P_4.

Round 4 ($\mathtt{bc}$): P_1 simply waits for incoming messages and remains silent.

Scenario 2: This scenario proceeds identically to Scenario 2 in Theorem 4 for the first three rounds, in particular in this scenario P_1 is honest with its input as σ and the adversary corrupts P_3 and P_4, who do the following:

Local Computation: Sample random strings r_3 and r_4 on behalf of P_3 and P_4 respectively. Compute first round messages $(\mathtt{bc}_3^{(1)}, \{\mathsf{msg}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_3^{(1)}((m_0', m_1'); r_3)$ and $(\mathtt{bc}_4^{(1)}, \{\mathsf{msg}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_4^{(1)}(r_4)$.
Sample another set of random strings $\overline{r_3}$ and $\overline{r_4}$ such that it is consistent with $\mathtt{bc}_3^{(1)}$ and $\mathtt{bc}_4^{(1)}$ respectively. More formally, suppose $(\overline{\mathtt{bc}}_3^{(1)}, \{\overline{\mathsf{msg}}_{3\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_3^{(1)}((m_0', m_1'); \overline{r_3})$ such that $\overline{\mathtt{bc}}_3^{(1)} = \mathtt{bc}_3^{(1)}$ and

$(\overline{\mathtt{bc}}_4^{(1)}, \{\overline{\mathsf{msg}}_{4\to j}^{(1)}\}_{j\in[5]}) \leftarrow \mathtt{nxt\text{-}msg}_4^{(1)}(\overline{r_4})$ such that $\overline{\mathtt{bc}}_4^{(1)} = \mathtt{bc}_4^{(1)}$.

Round 1 ($\mathtt{bc} + \mathtt{p2p}$): Broadcast $\mathtt{bc}_3^{(1)}$, send $\mathsf{msg}_{3\to j}^{(1)}$ to each P_j for $j \in \{2,5\}$, and send $\overline{\mathsf{msg}}_{3\to 1}^{(1)}$ to P_1. Similarly, on behalf of P_4, broadcast $\mathtt{bc}_4^{(1)}$, send $\mathsf{msg}_{4\to j}^{(1)}$ to each P_j for $j \in \{2,5\}$, and send $\overline{\mathsf{msg}}_{4\to 1}^{(1)}$ to P_1.
After receiving the first round messages, prepare a fake view V_3 comprising of the randomness r_3 and all received broadcast and point-to-point messages, except that $\mathsf{msg}_{1\to 3}^{(1)}$ received from P_1 in Round 1 is replaced with $\perp$. Similarly prepare a fake view V_4 comprising of the randomness r_4 and all received broadcast and point-to-point messages, except that $\mathsf{msg}_{1\to 4}^{(1)}$ received from P_1 in Round 1 is replaced with $\perp$. In other words, P_3 and P_4 pretend that they did not receive any point-to-point message from P_1 in the first round.

Round 2 ($\mathtt{bc}$): Behave honestly according to the fake view V_3 on behalf of P_3 and V_4 on behalf of P_4.

Round 3 ($\mathtt{p2p}$): Continue to behave honestly according to the fake view and the public broadcasts, except do not send a point-to-point message to P_1.

Round 4 ($\mathtt{bc}$). P_3 and P_4 both continue to behave honestly except that they pretend they did not receive any private communication from P_1 in Round 3, setting messages $\mathsf{msg}_{1\to 3}^{(3)}$ and $\mathsf{msg}_{1\to 4}^{(3)}$ to $\perp$.

Next, we make a sequence of claims:

Lemma 6. *In Scenario 1, the adversary obtains m_σ with overwhelming probability. Further, the output of P_2 and P_5 is m'_σ with overwhelming probability.*

Proof. First, we note that since P_1 and P_2 are both honest in Scenario 2, it must follow from G.O.D. security that P_1 in Scenario 2 outputs m_σ. Next, we note that the view of corrupt P_1's in Scenario 1 is identically distributed to the view of an honest P_1 in Scenario 2. Therefore, the adversary $\mathcal{A}_\sigma$ in Scenario 1 is able to compute m_σ (with probability at least $1-\epsilon$) as well. Lastly, it follows from G.O.D. security of Π_f that when Scenario 1 results in the adversary controlling P_1 to obtain m_σ (which occurs with overwhelming probability), then honest parties P_2 and P_5 must obtain m'_σ (with overwhelming probability as well).

Next, we proceed to argue that in the first two rounds of Π_f, the views of the honest parties at the end of the second round in Scenario 1 does not contain information about the input of P_1. We state and prove this in the lemma below.

Lemma 7. *Let $\mathsf{V}^\sigma_{2,3,4,5}$ denote the distributions of the combined views of $\{P_2, P_3, P_4, P_5\}$ at the end of Round 2 of the execution of Π_f in Scenario 1 (with respect to σ). Then, $\mathsf{V}^0_{2,3,4,5} \approx^\epsilon \mathsf{V}^1_{2,3,4,5}$ must hold.*

Proof. At a high level, this part of the proof follows a similar blueprint to that of Lemma 4 in Theorem 4. In particular, we can consider Scenario 3, which is very similar to Scenario 3 considered in Theorem 4 where parties P_1, P_3, and P_4 are honest, while P_2 and P_5 are corrupted. Notice that by the security of Π_f, P_2 and P_5 should not learn the secret input of honest P_1, namely σ (as it is not revealed by the output of f obtained by these parties). This must hold even if they are able to recreate in their head the view of P_3, P_4 in Scenario 1 at the end of the second round.

In Scenario 3 honest parties are P_1, P_3, and P_4 where P_1 uses input σ and randomness $\widetilde{r_1}$, while P_3 uses input (m'_0, m'_1) and randomness $\widetilde{r_3}$, and P_4 has no input and uses randomness $\widetilde{r_4}$. Moreover, P_2 and P_5 are corrupted but they behave honestly, except that at the end of Round 2 they do the following experiment in their head to emulate Scenario 1[8].

- Sample r_3^* such that it is consistent with the first round point-to-point messages $\mathsf{msg}_{3\to2}^{(1)}$ and $\mathsf{msg}_{3\to5}^{(1)}$ and first round broadcast message $\mathtt{bc}_3^{(1)}$ received from P_3. Similarly, sample r_4^* such that it is consistent with the first round point-to-point messages $\mathsf{msg}_{4\to2}^{(1)}$ and $\mathsf{msg}_{4\to5}^{(1)}$ and first round broadcast message $\mathtt{bc}_4^{(1)}$ received from P_4. More formally, r_3^* is such that $(\mathtt{bc}_3^{(1)}, \mathsf{msg}_{3\to1}^{*(1)}, \mathsf{msg}_{3\to2}^{(1)}, \mathsf{msg}_{3\to4}^{*(1)}, \mathsf{msg}_{3\to5}^{(1)}) \leftarrow \mathtt{nxt\text{-}msg}_3^{(1)}((m'_0, m'_1), r_3^*)$ and r_4^* is such that $(\mathtt{bc}_4^{(1)}, \mathsf{msg}_{4\to1}^{*(1)}, \mathsf{msg}_{4\to2}^{(1)}, \mathsf{msg}_{4\to3}^{*(1)}, \mathsf{msg}_{4\to5}^{(1)}) \leftarrow \mathtt{nxt\text{-}msg}_4^{(1)}(r_4^*)$.
- Compute $\mathtt{bc}_3^{*(2)} \leftarrow \mathtt{nxt\text{-}msg}_3^{(2)}((m'_0, m'_1), r_3^*, \{\mathtt{bc}_j^{(1)}\}_{j\in[5]}, \perp, \mathsf{msg}_{2\to3}^{(1)}, \mathsf{msg}_{4\to3}^{*(1)}, \mathsf{msg}_{5\to3}^{(1)})$ i.e. the second round broadcast message an honest P_3 with randomness r_3^* would compute, had it not received any private first-round message from P_1. Note that $\mathsf{msg}_{2\to3}^{(1)}$ and $\mathsf{msg}_{5\to3}^{(1)}$ are known to the adversary corrupting P_2 and P_5. Further $\mathsf{msg}_{4\to3}^{*(1)}$ was sampled by the adversary itself in the above step.
- Similarly, compute $\mathtt{bc}_4^{*(2)} \leftarrow \mathtt{nxt\text{-}msg}_4^{(2)}(r_4^*, \{\mathtt{bc}_j^{(1)}\}_{j\in[5]}\perp, \mathsf{msg}_{2\to4}^{(1)}, \mathsf{msg}_{3\to4}^{*(1)}, \mathsf{msg}_{5\to4}^{(1)})$ i.e. the second round broadcast message an honest P_4 with randomness r_4^* would compute had it not received any private first-round message from P_1. Note that $\mathsf{msg}_{2\to4}^{(1)}$ and $\mathsf{msg}_{5\to4}^{(1)}$ are known to the adversary corrupting P_2 and P_5. Further $\mathsf{msg}_{3\to4}^{*(1)}$ was sampled by the adversary itself in the above step.

Consider now the view V^* that at the end of the first 2 rounds, namely:

$\mathsf{V}^* = \{r_2, r_3^*, r_4^*, r_5, \{\mathtt{bc}_j^{(1)}\}_{j\in[5]}, \{\mathtt{bc}_j^{(2)}\}_{j\in\{1,2,5\}}, \{\mathtt{bc}_j^{*(2)}\}_{j\in\{3,4\}}, \{\mathsf{msg}_{j\to2}^{(1)}\}_{j\in[5]}, \{\mathsf{msg}_{j\to5}^{(1)}\}_{j\in[5]},$

$(\{\mathsf{msg}_{j\to3}^{(1)}\}_{j\in\{2,5\}}, \mathsf{msg}_{4\to3}^{*(1)}), (\{\mathsf{msg}_{j\to4}^{(1)}\}_{j\in\{2,5\}}, \mathsf{msg}_{3\to4}^{*(1)})\}$

We claim that V^* is identically distributed to the joint view of the honest parties at the end of the first 2 rounds in Scenario 1 (i.e. $\mathsf{V}_{2,3,4,5}^{\sigma}$) by noticing that:

- $\widetilde{r_i}$ $(i \in \{3,4\})$ in Scenario 3 is essentially analogous to $\overline{r_i}(i \in \{3,4\})$ in Scenario 1. Note that in both cases, P_1's second round messages are computed with respect to messages received based on this randomness.

– r_i^* ($i \in \{3,4\}$) in Scenario 3 is analogous to $r_i (i \in \{3,4\})$ in Scenario 1. Note that in both cases, this is the randomness with respect to which parties P_3 and P_4's messages are computed.

We observe that since P_1 (with input σ) is honest in Scenario 3, it follows from security of Π_f that the adversary corrupting P_2 and P_5 should not be able to create a view V^* that depends on the secret σ. Therefore, one can conclude that V^* must be independent of the secret σ. Finally, since V^* is distributed identically to $\mathsf{V}^{\sigma}_{2,3,4,5}$ as shown above, $\mathsf{V}^{0}_{2,3,4,5} \approx^{\epsilon} \mathsf{V}^{1}_{2,3,4,5}$ must hold.

Let us consider now **Scenario 1A** where the adversary $\mathcal{A}_\sigma$ behaves exactly as in Scenario 1 (using input σ), but additionally does not send the third round point-to-point message to P_5. We show that in Scenario 1A the parties P_2 and P_5 recover the output m'_σ. More formally, we prove the following Lemma.

Lemma 8. *The output of honest parties P_2 and P_5 in Scenario 1A is m'_σ with overwhelming probability.*

Proof. Since the protocol achieves G.O.D. and P_3 is honest, the output must be m'_b for some b. Suppose by contradiction that $b \neq \sigma$ with non-negligible probability. Then we consider the additional scenario *Scenario 1B*. Scenario 1B is identical to Scenario 1 except that P_5 is also corrupt but behaves honestly throughout. Then the adversary in Scenario 1B can obtain both m'_0 and m'_1 as follows: First, it obtains m'_σ (with overwhelming probability) as output of Scenario 1 on behalf of P_5 (as shown in Lemma 6). Next, in its head, the adversary emulates Scenario 1A by recomputing P_5's last round messages pretending that it did not receive any point-to-point message from P_1 in Round 3. This would allow the adversary to obtain a view that subsumes the view of honest P_5 in Scenario 1A, thereby learning $m_b = m_{1-\sigma}$ as well. This contradicts the security of Π_f – as the adversary in Scenario 1B managed to breach the privacy of honest P_3 by learning both m'_0 and m'_1 with non-negligible probability. We therefore conclude that $b = \sigma$ must hold and honest parties P_2 and P_5 recover the output m'_σ. This completes the proof of the lemma.

Let us now consider **Scenario 1C** where the adversary $\mathcal{A}_\sigma$ behaves exactly as in Scenario 1A (using input σ), but additionally corrupts P_2 who behaves honestly throughout the protocol. Note that in Scenario 1C, the honest parties do not receive any communication from P_1 after Round 2. We are now ready to make the following claim for Scenario 1C.

Lemma 9. *There exists no straight-line simulator $\mathcal{S}_{3,4,5}$ acting on behalf of honest parties $\{P_3, P_4, P_5\}$ in Scenario 1C which has more than $1/2 + 2\epsilon$ probability of extracting correctly the input σ of $\mathcal{A}_\sigma$.*

Proof. First, we recall that honest P_2 recovers m'_σ with overwhelming probability in Scenario 1A (Lemma 8). Since the corrupt P_2's view in Scenario 1C is identically distributed to honest P_2's view in Scenario 1A, we can conclude that the adversary in Scenario 1C is able to recover the output m'_σ with overwhelming

probability. From the above implication and the fact that $\mathcal{S}_{3,4,5}$ is straight-line, we can conclude that the input σ of $\mathcal{A}_\sigma$ needs to be extracted by the end of the Round 3 of Π_f.

Now, let us analyze the view of the simulator $\mathcal{S}_{3,4,5}$ at the end of Round 3, which is composed of (1) the combined view of honest parties $\{P_3, P_4, P_5\}$ in the end of the Round 2 (since honest parties are non-rushing and therefore their 3rd round messages is a deterministic functions of their randomness, inputs, and incoming messages until the 2nd round) and (2) the Round 3 message of corrupt P_2 (who behaves honestly throughout).

Noticing that in the first two rounds, Scenario 1C and Scenario 1 are actually the same, we can invoke Lemma 7 to claim that $\mathsf{V}^0_{3,4,5} \approx_\epsilon \mathsf{V}^1_{3,4,5}$, where $\mathsf{V}^\sigma_{3,4,5}$ indicates the distributions of the combined views of P_3, P_4, and P_5 in the end of the Round 2 when the input of P_1 is σ.

Moreover, let V^σ_2 denotes the random variable describing the view of honest P_2 at the end of Round 2 when the input of P_1 is σ. Note that corrupted P_2 behaves honestly throughout the protocol execution, in particular, she is non-rushing, and therefore her 3rd round depends only on the view of P_2 until the end of the Round 2 (and her input and randomness). Since by Lemma 7 it follows that $\mathsf{V}^0_2 \approx_\epsilon \mathsf{V}^1_2$, we can conclude that $\mathcal{S}_{3,4,5}$ has at most probability $1/2 + 2\epsilon$ to correctly extract the input σ of $\mathcal{A}_\sigma$ by the end of the Round 3. This completes the proof.

Finally, we can infer from Lemma 9 that $\mathcal{A}_\sigma$ induces a distribution of outputs in the ideal world which is distinguishable from that of the real world with non-negligible probability, which contradicts the security of Π_f; thereby concluding the proof.

Theorem 7. *There exists a function f such that no statistically secure n-party 4-round MPC protocol can compute f with G.O.D. against $2t < n \leq 2.5t$ corruptions, where the third round uses only point-to-point channels.*

Proof. This proof follows via the same player partition arguments explained for Theorem 5. In this case, we consider a 4-round MPC protocol π that works for n players where $n = 2.5t$ that does not use broadcast in the third round. This can be transformed into a 5-party MPC contradicting Theorem 6 by splitting n players into 5 disjoint subsets A_i for $i = 1, \ldots, 5$ and such that each A_i has size $n/5$. The claimed 5-party MPC will execute π by having P_i emulate the players in A_i in his head. Since π is secure for t corruptions, it is secure if players in any pair of subsets are corrupted, and this clearly means we contradict Theorem 6.

5 Upper Bound for MPC

In this section, we observe that the n-parties MPC protocol Π_{AKP} of [AKP23] for $\frac{n}{3} \geq t$ and $t > 1$ is optimal in terms of broadcast usage. We start by giving the following theorem, which is similar in spirit to Theorem 7 of [DMR+21].

Theorem 8. *Let Π be a 4-round MPC protocol that computes the functionality $\mathcal{F}$ with guaranteed output delivery, which uses broadcast in the last round. If Π admits a simulator that can extract inputs from the first three rounds, then Π achieves the same guarantee when the last round is run over point-to-point channels.*

The formal argument is provided in the full version of the paper [CDR+25].

References

AKP20a. Benny Applebaum, Eliran Kachlon, and Arpita Patra. The resiliency of MPC with low interaction: The benefit of making errors (extended abstract). In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 562–594, Durham, NC, USA, November 16–19, 2020.

AKP20b. Benny Applebaum, Eliran Kachlon, and Arpita Patra. The round complexity of perfect MPC with active security and optimal resiliency. In *61st Annual Symposium on Foundations of Computer Science*, pages 1277–1284, Durham, NC, USA, November 16–19, 2020. IEEE Computer Society Press.

AKP23. Benny Applebaum, Eliran Kachlon, and Arpita Patra. The round complexity of statistical MPC with optimal resiliency. pages 1527–1536. ACM Press, 2023.

AL11. Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly-secure multiparty computation. Cryptology ePrint Archive, Report 2011/136, 2011.

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

CDR+23. Michele Ciampi, Ivan Damgård, Divya Ravi, Luisa Siniscalchi, Yu Xia, and Sophia Yakoubov. Broadcast-optimal four-round MPC in the plain model. In *TCC 2023: 21st Theory of Cryptography Conference, Part II*, Lecture Notes in Computer Science, pages 3–32, November 2023.

CDR+25. Michele Ciampi, Ivan Damgård, Divya Ravi, Luisa Siniscalchi, and Sophia Yakoubov. Information-theoretic broadcast-optimal MPC. Cryptology ePrint Archive, Paper 2025/1710, 2025.

CGZ20. Ran Cohen, Juan A. Garay, and Vassilis Zikas. Broadcast-optimal two-round MPC. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 828–858, Zagreb, Croatia, May 10–14, 2020.

DMR+21. Ivan Damgård, Bernardo Magri, Divya Ravi, Luisa Siniscalchi, and Sophia Yakoubov. Broadcast-optimal two round MPC with an honest majority. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 155–184, Virtual Event, August 16–20, 2021.

DRSY22. Ivan Damgård, Divya Ravi, Luisa Siniscalchi, and Sophia Yakoubov. Minimizing setup in broadcast-optimal two round mpc. Cryptology ePrint Archive, Paper 2022/293, 2022. To appear in Eurocrypt2023.

DRSY23. Ivan Damgård, Divya Ravi, Luisa Siniscalchi, and Sophia Yakoubov. Broadcast-optimal two round MPC with asynchronous peer-to-peer channels. Lecture Notes in Computer Science, pages 87–106, 2023.

GGHR14. Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 74–94, San Diego, CA, USA, February 24–26, 2014.

GGOR12. Juan Garay, Clint Givens, Rafail Ostrovsky, and Pavel Raykov. Broadcast (and round) efficient verifiable secret sharing. Cryptology ePrint Archive, Report 2012/130, 2012.

GGOR14. Juan A. Garay, Clint Givens, Rafail Ostrovsky, and Pavel Raykov. Broadcast (and round) efficient verifiable secret sharing. In Carles Padró, editor, *ICITS 13: 7th International Conference on Information Theoretic Security*, volume 8317 of *Lecture Notes in Computer Science*, pages 200–219, Singapore, 2014.

GIKR01. Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. The round complexity of verifiable secret sharing and secure multicast. In *33rd Annual ACM Symposium on Theory of Computing*, pages 580–589, Crete, Greece, July 6–8, 2001. ACM Press.

Gol04. Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

HLP11. Shai Halevi, Yehuda Lindell, and Benny Pinkas. Secure computation on the web: Computing without simultaneous interaction. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 132–150, Santa Barbara, CA, USA, August 14–18, 2011.

KPR10. Ranjit Kumaresan, Arpita Patra, and C. Pandu Rangan. The round complexity of verifiable secret sharing: The statistical case. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 431–447, Singapore, December 5–9, 2010.

PCRR09. Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 487–504, Santa Barbara, CA, USA, August 16–20, 2009.