A New Approach to Round-Optimal Secure Multiparty Computation

Prabhanjan Ananth
University of California Los Angeles

Arka Rai Choudhuri
Johns Hopkins University

Abhishek Jain
Johns Hopkins University

Abstract

We present a new approach towards constructing round-optimal secure multiparty computation (MPC) protocols against malicious adversaries without trusted setup assumptions. Our approach builds on ideas previously developed in the context of covert multiparty computation [Chandran et al., FOCS'07] even though we do not seek covert security. Using our new approach, we obtain the following results:

- A five round MPC protocol based on the Decisional Diffie-Hellman (DDH) assumption.
- A four round MPC protocol based on one-way permutations and sub-exponentially secure DDH. This result is *optimal* in the number of rounds.

Previously, no four-round MPC protocol for general functions was known and five-round protocols were only known based on indistinguishability obfuscation (and some additional assumptions) [Garg et al., EUROCRYPT'16].

1 Introduction

The notion of secure multiparty computation (MPC) [Yao86, GMW87] is fundamental in cryptography. Informally speaking, an MPC protocol allows mutually distrusting parties to jointly evaluate a function on their private inputs in such a manner that the protocol execution does not leak anything beyond the output of the function.

A fundamental measure of efficiency in MPC is round complexity, i.e., the number of rounds of communication between the parties. The round complexity of MPC has been extensively studied over the last three decades. Protocols with smaller round complexity are more desirable so as to minimize the effect of network latency, which in turn decreases the time complexity of the protocol.

In this work, we study round-optimal MPC against malicious adversaries who may corrupt an arbitrary subset of parties, in the plain model without any trusted setup assumptions. We consider the traditional simultaneous message model for MPC, where in each round of the protocol, each party simultaneously broadcasts a message to the other parties.

A lower bound for this setting was established last year by Garg et al. [GMPP16] who proved that three rounds are insufficient for coin-tossing w.r.t. black-box simulation. (Their work builds on [KO04] who proved the necessity of five rounds for coin-tossing in the unidirectional message model.) In the positive direction, several constant-round MPC protocols were constructed in a long sequence of works, based on a variety of assumptions and techniques (see, e.g., [KOS03, Pas04, PW10, Wee10, Goy11]). Garg et al. [GMPP16] established an upper bound on the exact round complexity of MPC by constructing a *five* round protocol based on indistinguishability obfuscation [BGI$^+$01, GGH$^+$13] and some additional assumptions.[1] Their work constitutes the state of the art on this subject.

Our Goals. Presently, no constructions of indistinguishability obfuscation are known from standard assumptions. This motivates the following important question:

Does there exist a five round maliciously-secure MPC protocol for general functions based on standard polynomial-time assumptions?

Furthermore, given the gap between the lower bound (three rounds) and the upper bound (five rounds) established by [GMPP16], we ask whether their upper bound is tight:

Does there exist a four round maliciously-secure MPC protocol for general functions?

In this work, we resolve both of these questions in the affirmative.

The Main Barrier. We highlight the main conceptual barrier towards achieving our goals. Garg et al. [GMPP16] follow a natural two-step approach to obtain their positive results: in the first step, they construct a four round multiparty coin-tossing protocol. In the next step, they use their coin-tossing protocol to replace the common random string (CRS) in a two-round MPC protocol in the CRS model [GGHR14, MW16].

We note, however, that this approach, in general, cannot do better than five rounds. Indeed, since at least one of the rounds of the two-round MPC must depend upon the CRS, we can only hope to parallelize its first round with the coin-tossing protocol. Since coin-tossing requires four rounds, this only yields a five round protocol at best.

A New Approach. In this work, we present a new approach towards constructing round-optimal MPC protocols in the plain model. At a high level, our approach implements the classical GMW methodology [GMW87] for constructing maliciously-secure MPC protocols, *with a crucial twist*, to minimize the number of rounds. This approach is inspired by the beautiful work of Chandran et al. [CGOS07] for constructing covert multiparty computation protocols [vHL05, CGOS07, GJ10].

Recall that the GMW compiler transforms a semi-honest MPC protocol into a maliciously secure one by requiring the parties to prove (using zero-knowledge proofs [GMR85]) that each message in the semi-honest protocol was computed "honestly." Towards our goal of minimizing round complexity, we cannot afford to prove honest behavior with every round of semi-honest MPC. Therefore, in our approach, the parties prove honest behavior only *once*.

At first, such an approach may sound completely absurd. If each party is only required to give a single proof of honest behavior, then a malicious adversary may choose to cheat in the first few rounds of the semi-honest MPC protocol. By the time the proof is completed and the honest parties are able to detect cheating, it may already be "too late." Indeed, the opportunity to cheat in even a single round may be sufficient for a malicious adversary to completely break the security of a semi-honest protocol. Therefore, it is not at all clear why such an approach can be implemented in a secure manner.

In order to tackle this problem, we design a "special-purpose" semi-honest MPC protocol that remains partially immune to malicious behavior before the last round of the protocol. Specifically, in such a protocol, an adversary can influence the protocol outcome but not learn any private information by behaving maliciously before the last round. We then "shield" the last round from being revealed to the adversary until it has proven honest behavior for all of the preceding rounds. A single proof suffices to accomplish this task. By parallelizing this proof with the semi-honest MPC, we are able to minimize the round complexity.

We note that the above idea of delaying the proof of honest behavior to the end of the computation was first developed in [CGOS07]. While they developed this technique to achieve covert security (namely, hiding protocol participation from other players), we use it in our setting to minimize round complexity.

[1] Garg et al. also construct a four-round protocol for the coin-tossing functionality. In this work, we are interested in MPC for general functions.

1.1 Our Results

We present a new approach for constructing round-efficient MPC protocols that are secure against malicious adversaries in the plain model. Using this approach, we are able to achieve both of our aforementioned goals.

I. Robust Semi-honest MPC. As a first step towards obtaining our results for maliciously-secure MPC, we construct a four round *robust* semi-honest MPC protocol that remains partially immune to malicious behavior. In this protocol, at the end of the first three rounds of computation, each party receives a secret share of the function output. In the last round, the parties simply exchange their shares to reconstruct the output. The key security property of this protocol is that if the adversary cheats in the first three rounds, then it can only influence the function output, but not learn any private information.

We construct such an MPC scheme for general functions assuming the existence of low-depth pseudorandom generators (PRGs) and a two-round "covert" oblivious transfer (OT) protocol [vHL05].[2] Both of these primitives can be instantiated from the Decisional Diffie-Hellman (DDH) assumption.

Theorem 1. *Assuming DDH, there exists a four round robust semi-honest MPC protocol for general functions.*

The above result may be of independent interest.

II. Maliciously-secure MPC. Using theorem 1, we next construct maliciously-secure MPC protocols in the plain model. Our first result is a five round MPC protocol based on any four-round robust semi-honest MPC, injective one-way functions and collision-resistant hash functions (CRHFs).

Theorem 2 (Five Rounds)**.** *Assuming DDH, there exists a five round maliciously-secure MPC protocol for computing general functions.*

We next modify our five round protocol to obtain a four round protocol, albeit using sub-exponential hardness. The security of our construction uses complexity leveraging between multiple primitives.

Theorem 3 (Four Rounds)**.** *Assuming one-way permutations and sub-exponentially secure DDH, there exists a four round maliciously-secure MPC protocol for computing general functions.*

1.2 Our Techniques

As discussed earlier, the approach of Garg et al. [GMPP16] for constructing maliciously-secure MPC protocols is unsuitable for achieving our goals. Therefore, we develop a new approach for constructing round-efficient MPC against malicious adversaries.

At a high-level, our approach implements the GMW paradigm for constructing maliciously-secure MPC protocols, with a crucial twist. Recall that the GMW paradigm transforms a semi-honest MPC protocol into a maliciously secure one using the following three steps: (1) first, the parties commit to their inputs and random tapes. (2) Next, the parties perform coin-tossing to establish an unbiased random tape for each party. (3) Finally, the parties run the semi-honest MPC protocol where along with every message, each party also gives zero-knowledge proof of "honest" behavior consistent with the committed input and random tape.

Both steps (2) and (3) above introduce additional rounds of interaction, and constitute the main bottleneck towards constructing round-optimal MPC.

Main Ideas. Towards this, we develop two key modifications to the GMW compiler:

1. **"One-shot" proof**: Instead of requiring the parties to give a proof of honest behavior in each round of the underlying semi-honest protocol, we use a "delayed verification" technique where the

[2] We use low-depth PRGs to obtain degree-three randomizing polynomials for general functions [AIK06].

parties prove honest behavior only *once*, towards the end of the protocol. As we explain below, this allows us to limit the overhead of additional rounds introduced by zero-knowledge proofs in the GMW compiler.

The idea of delayed verification was previously developed in the work of Chandran et. al. [CGOS07]. Interestingly, while they used this technique to achieve security in the setting of covert computation [vHL05, CGOS07], we use this technique to minimize the round complexity of our protocol.

2. **No coin tossing**: Second, we eliminate the coin-tossing step (i.e., step 2). Note that by removing coin-tossing, we implicitly allow the adversarial parties to potentially use "bad" randomness in the protocol. To ensure security in this scenario, we will use a special semi-honest MPC protocol that is secure against bad randomness. This idea has previously been used in many works (see, e.g.,[AJL$^+$12, MW16]).

We now elaborate on the first step, which constitutes the conceptual core of our work. We consider semi-honest MPC protocols with a specific structure consisting of two phases: (a) *Computation phase*: in the first phase of the protocol, the parties compute the function such that each party obtains a secret-share of the output. (b) *Output phase*: In the second phase, the parties exchange their output shares with each other to compute the final output. This phase consists of only one round and is deterministic. Note that standard MPC protocols such as [GMW87] follow this structure.

At a high-level, we implement our delayed verification strategy as follows: the parties first run the computation phase of the semi-honest protocol "as is" without giving any proofs. At the end of this phase, each party gives a single proof that it behaved honestly throughout the computation phase (using the committed input and random tape). If all the proofs verify, then the parties execute the output phase.

Right away, one may notice a glaring problem in the above approach. If the computation phase is executed without any proof of honest behavior, the adversary may behave maliciously in this phase and potentially learn the honest party inputs even before the output phase begins! Indeed, standard semi-honest MPC protocols do not guarantee security in such a setting.

To combat this problem, we develop a special purpose semi-honest MPC protocol that remains "partially immune" to malicious behavior. Specifically, such a protocol maintains privacy against malicious adversaries *until the end of the computation phase*. However, output correctness is not guaranteed if the adversary behaved maliciously in the computation phase. We refer to such an MPC protocol as *robust* semi-honest MPC. Later, we describe a four-round construction of robust semi-honest MPC where the first three rounds correspond to the computation phase and the last round constitutes the output phase.

Note that the robustness property as described above perfectly suits our requirements because in our compiled protocol, the output phase is executed only after each party has proven that it behaved honestly during the computation phase. This ensures full security of our compiled protocol.

A New Template for Malicious MPC. Putting the above ideas together, we obtain the following new template for maliciously-secure MPC:

- First, each party commits to its input and randomness using both a three-round extractable commitment scheme[3], and a non-interactive commitment scheme. In parallel, the parties also execute the computation phase of a four-round robust semi-honest MPC.
- Next, each party proves to every other party that it behaved honestly during the first three rounds.
- Finally, the parties execute the output phase of the robust semi-honest MPC and once again prove that their message is honestly computed.

[3] We use a variant of the extractable commitment scheme in [Ros04] for this purpose. This variant has been used in many prior works such as [GJO10, GGJS12, Goy12] because it is "rewinding secure" – a property that is used in the security proofs.

In order to obtain a five round protocol from this template, we need to parallelize the proofs with the other protocol messages. For this purpose, we use delayed-input proofs [LS90] where the instance is only required in the last round.[4] In particular, we use four-round delayed input zero-knowledge (ZK) proofs whose first three messages are executed in parallel with the first three rounds of the robust semi-honest MPC. This yields us a five round protocol.

We remark that during simulation, our simulator is able to extract the adversary's input only at the end of the third round. This means that we need to simulate the first three rounds of the robust semi-honest MPC without knowledge of the adversary's input (or the function output). Our robust semi-honest MPC satisfies this property; namely, the simulator for our robust semi-honest MPC needs the adversary's input and randomness (and the function output) only to simulate the output phase.

Four Rounds: Main Ideas. We next turn to the problem of constructing four-round MPC. At first, it is not clear how to obtain a four round protocol using the above template. Indeed, as argued earlier, we cannot afford to execute the output phase without verifying that the parties behaved honestly during the computation phase. In the above template, the output phase is executed *after* this verification is completed. Since three-round zero-knowledge proofs with polynomial-time simulation are not known presently, the verification process in the above protocol requires four rounds. Therefore, it may seem that that we are limited to a five round protocol.

Towards that, we note that our robust semi-honest MPC (described later) satisfies the following property: in order to simulate the view of the adversary (w.r.t. the correct output), the simulator only needs to "cheat" in the output phase (i.e., the last round). In particular, the simulation of the computation phase can be done "honestly" using random inputs for the honest parties. In this case, we do not need full-fledged ZK proofs to establish honest behavior in the computation phase; instead, we only need *strong* witness indistinguishable (WI) proofs. Recall that in a strong WI proof system, for any two indistinguishable instance distributions D_1 and D_2, a proof for $x_1 \leftarrow D_1$ using a witness w_1 is indistinguishable from a proof for $x_2 \leftarrow D_2$ using a witness w_2. This suffices for us because using strong WI, we can switch from an honest execution of the computation phase using the real inputs of the honest parties to another honest execution of the computation phase using random inputs for the honest parties.

Recently, Jain et al. [JKKR17] constructed three-round delayed-input strong WI proofs of knowledge from the DDH assumption. However, their proof system only guarantees strong WI property if the entire statement is chosen by the prover in the last round. In our case, this is unfortunately not true, and hence we cannot use their construction. Therefore, we take a different route, albeit at the cost of sub-exponential hardness assumptions. Specifically, we observe that by relying upon sub-exponential hardness, we can easily construct a three-round (delayed-input) strong WI argument by combining any three-round (delayed-input) WI proof of knowledge with a one or two-message "trapdoor phase" in our simultaneous message setting. For example, let f be a one-way permutation. The trapdoor phase can be implemented by having the verifier send $y = f(x)$ for a random x in parallel with the first prover message. The statement of the WI proof of knowledge is changed to: either the original statement is true or the prover knows x.

Now, by running in exponential time in the hybrids, we can break the one-way permutation to recover x and then prove knowledge of x. This allows us to switch from honest execution of the computation phase using the real inputs of the honest parties to another honest execution using random inputs. After this switch, we can go back to proving the honest statement which can be done in polynomial time. This ensures that our final simulator is also polynomial time.

Handling Non-malleability Issues. So far, we ignored non-malleability related issues in our discussion. However, as noted in many prior works, zero-knowledge proofs with standard soundness guarantee do not suffice in the setting of constant-round MPC. Indeed, since proofs are being executed in parallel, we

[4] Note that the witness for these proofs corresponds to the adversary's input and random tape which is already fixed in the first round.

need to ensure that an adversary's proofs remain sound even when the honest party's proofs are being simulated [Sah99].

We handle such malleability issues by using the techniques developed in a large body of prior works. In our five round MPC protocol, we make non-black-box use of (a slight variant) of the four-round non-malleable zero-knowledge (NMZK) argument of [COSV17] to ensure that adversary's proofs remain sound even during simulation. More specifically, following prior works such as [BPS06, GJO10, GGJS12, Goy12], we establish a "soundness lemma" to ensure that the adversary is behaving honestly across the hybrids. We use the extractability property of the non-malleable commitment used inside the non-malleable zero-knowledge argument to prove this property.

In our four round protocol, we use the above NMZK to prove honest behavior in the output phase. In order to prove honest behavior in the computation phase, we use a slightly modified version of the strong WI argument system described above which additionally uses the two-round extractable non-malleable commitment scheme of [KS17] to achieve the desired non-malleability properties.[5] Unlike the five round construction, here, we rely upon complexity leveraging in several of the hybrids to argue the "soundness lemma" as well as to tackle some delicate rewinding-related issues that are commonplace in such proofs. We refer the reader to the technical sections for details.

Robust Semi-honest MPC. We now briefly describe the high-level ideas in our four-round construction of robust semi-honest MPC for general functionalities. Towards this, we note that it suffices to achieve a simpler goal of constructing robust semi-honest MPC for a restricted class of functionalities, namely, for computing randomized encodings.[6] That is, in order to construct a robust MPC for a n-party functionality F, it suffices to construct a robust MPC for a n-functionality F_{rnd} that takes as input $(x_1, r_1; \cdots ; x_n, r_n)$ and outputs a randomized encoding of $F(x_1, \ldots, x_n)$ using randomness $r_1 \oplus \cdots \oplus r_n$. This is because all the parties can jointly execute the protocol for F_{rnd} to obtain the randomized encoding. Each party can then individually execute the decoding algorithm of the randomized encoding to recover the output $F(x_1, \ldots, x_n)$. Note that this transformation preserves round complexity.

To construct a robust semi-honest n-party protocol for F_{rnd}, we consider a specific type of randomized encoding defined in [AIK06]. In particular, they construct a degree 3 randomizing polynomials [7] for arbitrary functionalities based on low-depth pseudorandom generators. In their construction, every output bit of the encoding can be computed by a degree 3 polynomial on the input and the randomness. Hence, we further break down the goal of constructing a protocol for F_{rnd} into the following steps:

- Step 1: Construct a robust semi-honest MPC 3-party protocol for computing degree 3 terms. In particular, at the end of the protocol, every party who participated in the protocol get a secret share $x_1 x_2 x_3$, where x_q is the q^{th} party's input for $q \in \{1, 2, 3\}$. The randomness for the secret sharing comes from the parties in the protocol.
- Step 2: Using Step 1, construct a robust semi-honest MPC protocol to compute degree 3 polynomials.
- Step 3: Using Step 2, construct a robust semi-honest MPC protocol for F_{rnd}.

Steps 2 and 3 can be achieved using standard transformations and these transformations are round preserving. Thus, it suffices to achieve Step 1 in four rounds. Suppose P_1, P_2 and P_3 participate in the protocol. Roughly, the protocol proceeds as follows: P_1 and P_2 perform a two message covert OT protocol to receive a share of $x_1 x_2$. Then, P_1 and P_3 perform a two message OT protocol to receive a share of $x_1 x_2 x_3$. We need to do more work to ensure that at the end, all of them have shares of $x_1 x_2 x_3$. Further,

[5] Our security proof can be adapted to instead work with a three-round delayed-input public-coin non-malleable commitment scheme [GPR16, COSV16]. See Appendix A.4 for further discussion.

[6] A randomized encoding of function f and input x is such that, the output $f(x)$ can be recovered from this encoding and at the same time, this encoding should not leak any information about either f or x.

[7] The terms randomized encodings and randomizing polynomials are interchangeably used.

the robustness guarantee is argued using the covert security of the OT protocol. We refer the reader to the technical sections for more details.

1.3 Concurrent Work

In a concurrent and independent work, Brakerski, Halevi and Polychroniadou [BHP17] construct a maliciously-secure 4-round MPC protocol based on the sub-exponential hardness of the Learning with Errors (LWE) problem and on the adaptive commitments of [PPV08]. Their approach is very different from ours, most notably in the initial step, in that they construct and use a 3-round protocol against semi-malicious adversaries from LWE, while we construct and use a robust semi-honest MPC protocol from DDH.

1.4 Related Work

The study of constant-round protocols for MPC was initiated by Beaver et al. [BMR90]. They constructed constant-round MPC protocols in the presence of honest majority. Subsequently, a long sequence of works constructed constant-round MPC protocols against dishonest majority based on a variety of assumptions and techniques (see, e.g., [KOS03, Pas04, PW10, Wee10, Goy11]). Very recently, Garg et al. [GMPP16] constructed five round MPC using indistinguishability obfuscation and three-round robust non-malleable commitments. They also construct a six-round MPC protocol using learning with errors (LWE) assumption and three-round robust non-malleable commitments. All of these results are in the plain model where no trusted setup assumptions are available.

Asharov et. al. [AJL+12] constructed three round MPC protocols in the CRS model. Subsequently, two-round MPC protocols in the CRS model were constructed by Garg et al. [GGHR14] using indistinguishability obfuscation, and by Mukherjee and Wichs [MW16] using LWE assumption.

2 Preliminaries

For the definitions of all the underlying primitives used in our constructions, we refer the reader to Appendix A. Below, we provide the definition of robust semi-honest MPC.

Robust Semi-Honest MPC. We consider semi-honest secure multi-party computation protocols that satisfy an additional *robustness* property. Intuitively the property says that, except the final round, the messages of honest parties reveal no information about their inputs even if the adversarial parties behave *maliciously.*

Definition 1. *Let F be an n-party functionality. Let $\mathcal{A} = (\mathcal{A}^1, \mathcal{A}^2)$ represent a PPT algorithm controlling a set of parties $S \subseteq [n]$. For a t-round protocol computing F, we let $\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x}, z)$ denote the view of $\mathcal{A}^1$ during the first $t-1$ rounds in the real execution of the protocol on input $\vec{x} = (x_1, \cdots, x_n)$ and auxiliary input z. We require that at the end of the first $t-1$ rounds in the real protocol, $\mathcal{A}^1$ outputs* state *and* $(\mathsf{inp}, \mathsf{rand})$ *on a special tape where either* $(\mathsf{inp}, \mathsf{rand}) = (\perp, \perp)$ *(if $\mathcal{A}^1$ behaved maliciously) or* $(\mathsf{inp}, \mathsf{rand}) = (\{\widehat{x}_i\}_{i \in S}, \{\widehat{r}_i\}_{i \in S})$ *which is consistent with the honest behavior for* $\mathsf{RealExec}_{(t-1)}$ *(first $t-1$ rounds).*

*A protocol is said to be a "**robust**" secure multiparty computation protocol for F if for every PPT adversary $\mathcal{A} = (\mathcal{A}^1, \mathcal{A}^2)$ controlling a set of parties S in the real world, where $\mathcal{A}^2$ is semi-honest, there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}^1, \mathsf{Sim}^2)$ *such that for every initial input vector $\vec{x}$, every auxiliary input z*

- *If* $(\mathsf{inp},\mathsf{rand}) \neq (\bot,\bot)$, *then:*

$$\left(\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x},z),\ \mathsf{RealExec}^{\mathcal{A}^2}_{t}(\vec{x},\mathsf{state})\right) \approx_c \left(\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x},z),\ \mathsf{Sim}^2(\{\widehat{x_i}\}_{i\in S},\{\widehat{r_i}\}_{i\in S},y,\mathsf{state})\right)$$
$$\approx_c \left(\mathsf{Sim}^1(z),\ \mathsf{Sim}^2\left(\{\widehat{x_i}\}_{i\in S},\{\widehat{r_i}\}_{i\in S},y,\mathsf{state}\right)\right).$$

Here $y = F(\widehat{x_1},\ldots,\widehat{x_n})$, *where* $\widehat{x_i} = x_i$ *for* $i \notin S$. *And* $\mathsf{RealExec}^{\mathcal{A}^2}_{t}(\vec{x},\mathsf{state})$ *is the view of adversary* $\mathcal{A}^2$ *in the* t^{th} *round of the real protocol.*

- *Else,*

$$\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x},z) \approx_c \mathsf{Sim}^1(z).$$

Note that, in general, a semi-honest MPC protocol may not satisfy this property. In section 3, we construct a four-round semi-honest MPC protocol with robustness property.

Remark 1. *Our definition of robust semi-honest MPC implies semi-malicious security. Informally, a semi-malicious adversary is one follows protocol instructions but may choose its randomness arbitrarily. See [AJL+12] for a formal definition.*

3 Four Round Robust Semi-Honest MPC

We first describe the tools required for our construction. We require,

- Two message 1-out-of-2 covert oblivious transfer protocol (Theorem 11). Denote this by OT.
- Degree 3 randomizing polynomials for arbitrary polynomial sized circuits (Theorem 12). Denote this by $\mathsf{RP} = (\mathsf{CktE},\mathsf{D})$.

Both the tools mentioned above can be instantiated from DDH.

Construction. Our goal is to construct an n-party MPC protocol Π^F_{sh} secure against semi-honest adversaries for an n-party functionality F. Moreover, we show that Π^F_{sh} satisfies Robust property (Definition 1). We employ the following steps:

- **Step I:** We first construct an 3-party semi-honest MPC protocol $\Pi^{\mathsf{3MULT}}_{\mathsf{sh}}$ for the functionality $\mathsf{3MULT}$ defined below. This protocol is a three round protocol. However, we view this as a four round protocol (with the last round being empty) – the reason behind doing this is because this protocol will be used as a sub-protocol in the next steps and in the proof, the programming of the simulator occurs only in the fourth round.

$$\mathsf{3MULT}((x_1,r_1);(x_2,r_2);(x_3)) \text{ outputs } (r_1;\ r_2;\ x_1x_2x_3+r_1+r_2)$$

- **Step II:** We use $\Pi^{\mathsf{3MULT}}_{\mathsf{sh}}$ to construct an n-party semi-honest MPC protocol $\Pi^{\mathsf{3POLY}\{p\}}_{\mathsf{sh}}$ for the functionality $\mathsf{3POLY}\{p\}$ defined below, where p is a degree 3 polynomial in $\mathbb{F}_2[\mathbf{y}_1,\ldots,\mathbf{y}_N]$. This protocol is a four round protocol and it satisfies robust property.

$$\mathsf{3POLY}\{p\}(X_1;\cdots;X_n) \text{ outputs } p(\mathbf{y}_1,\ldots,\mathbf{y}_N),$$

  where $X_1,\ldots,X_n$ are partitions of $\mathbf{y}_1,\ldots,\mathbf{y}_N$.

- **Step III:** We use $\Pi^{\mathsf{3POLY}}_{\mathsf{sh}}$ to construct an n-party semi-honest MPC protocol Π^F_{sh}. This protocol is a four round protocol and it satisfies robust property.

We now describe the steps in detail.

Step I: Constructing $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$. Denote the parties by P_1, P_2 and P_3. Denote the input of P_1 to be (x_1, r_1), the input of P_2 to be (x_2, r_2) and the input of P_3 to be (x_3). The protocol works as follows:

- **Round 1**: P_1 participates in a 1-out-of-2 oblivious transfer protocol OT_{12} with P_2. P_1 plays the role of receiver. It generates the first message of OT_{12} as a function of x_1.

  Simultaneously, P_2 and P_3 participate in a 1-out-of-2 protocol OT_{23}. P_3 takes the role of the receiver. It generates the first message of OT_{23} as a function of x_3.

- **Round 2**: P_2 sends the second message in OT_{12} as a function of $(x_2 \cdot 0 + r_2';\ x_2 \cdot 1 + r_2')$, where r_2' is sampled at random. P_2 sends the second message in OT_{23} as a function of $(0 \cdot r_2' + r_2;\ 1 \cdot r_2' + r_2)$.

  Simultaneously, P_1 and P_3 participate in a OT protocol OT_{13}. P_3 takes the role of the receiver. It sends the first message of OT_{13} as a function of x_3.

- **Round 3**: Let u be the value recovered by P_1 from OT_{12}. P_1 sends the second message to P_3 in OT_{13} as a function of $(u \cdot 0 + r_1, u \cdot 1 + r_1)$. Let α_3' recovered from OT_{13} by P_3 and let α_3'' be the output recovered from OT_{23}.

P_1 outputs $\alpha_1 = r_1$, P_2 outputs $\alpha_2 = r_2$ and P_3 outputs $\alpha_3 = \alpha_3' + \alpha_3''$ (operations performed over $\mathbb{F}_2$).

Theorem 4. *Assuming the correctness of* OT, $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ *satisfies correctness property.*

The proof can be found in B.1.

Theorem 5. *Assuming the security of* OT, $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ *is a robust semi-honest three-party secure computation protocol satisfying Definition 1.*

The proof can be found in B.2.

Step II: Constructing $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$. We first introduce some notation. Consider a polynomial $q \in \mathbb{F}_2[\mathbf{y}_1, \ldots, \mathbf{y}_N]$ with coefficients over $\mathbb{F}_2$. We define the set $\mathsf{MonS}\{q\}$ as follows: a term $t \in \mathsf{MonS}\{q\}$ if and only if t appears in the expansion of the polynomial q. We define $\mathsf{MonS}\{q\}_i$ as follows: a term $t \in \mathsf{MonS}\{q\}_i$ if and only if $t \in \mathsf{MonS}\{q\}$ and t contains the variable $\mathbf{y}_i$.

We now describe $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$.

Protocol $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$: Let $P_1, \ldots, P_n$ be the set of parties in the protocol. Let X_i be the input set of P_i for every $i \in [n]$. We have, $\sum_{i=1}^{n} |X_i| = N$ and $X_i \cap X_j = \emptyset$ for $i \neq j$. Every $x \in X_i$ corresponds to a unique variable $\mathbf{y}_j$ for some j.

- For every $i \in [n]$, party P_i generates n additive shares $s_{i,1}, \ldots, s_{i,n}$ of 0. It sends share $s_{i,j}$ to P_j in the first round.

- In parallel, for every term t in the expansion of p, do the following:

  - If t is of the form x_i^3, then P_i computes x_i^3.
  - If t is of the form $x_i^2 x_j$ then pick $k \in [n]$ and $k \neq i, k \neq j$. Let r_i^t and r_j^t be the randomness, associated with t, sampled by P_i and P_j respectively. The parties $P_i(x_i, r_i^t)$, $P_j(x_j, r_j^t)$ and $P_k(1)$ execute $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ to obtain the corresponding shares α_i^t, α_j^t and α_k^t. Note that this finishes in the third round.
  - If t is of the form $x_i x_j x_k$, then parties P_i, P_j and P_k sample randomness r_i^t, r_t^j and r_k^t respectively. Then, they execute $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ on inputs (x_i, r_i^t), (x_j, r_j^t) and (x_k) to obtain the corresponding shares α_i^t, α_j^t and α_k^t. Note that this finishes in the third round.

- After the third round, P_i adds all the shares he has so far (including his own shares) and he broadcasts his final share s_i to all the parties. This consumes one round.
- Finally, P_i outputs $\sum_{i=1}^{n} s_i$.

Theorem 6. *Assuming $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ satisfies correctness, $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$ satisfies correctness property.*

The proof can be found in B.3.

Theorem 7. *Assuming the security of $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$, $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$ is a robust semi-honest MPC protcol satisfying Definition 1 as long as $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ satisfies Definition 1.*

The proof can be found in B.4.

Step III: Constructing Π_{sh}^{F}. We describe Π_{sh}^{F} below.

Protocol Π_{sh}^{F}: Let C be a circuit representing F. That is, $F(x_1;\ldots,x_n) = C(x_1||\cdots||x_n)$. Let $\mathsf{RP.CktE}(C) = (p_1,\ldots,p_m)$. Note that p_i, for every i, is a degree 3 polynomial in $\mathbb{F}_2[\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{r}_1,\ldots,\mathbf{r}_N]$. Construct polynomial $\hat{p_i} \in \mathbb{F}_2[\mathbf{y}_1,\ldots,\mathbf{y}_n,,\mathbf{r}_{1,1},\ldots,\mathbf{r}_{n,N}]$ by replacing $\mathbf{r}_j$, for every $j \in [N]$, in p_i by the polynomial $\sum_{k=1}^{n} \mathbf{r}_{k,j}$. Note that $\hat{p_i}$ is still a degree 3 polynomial.

P_i samples randomness $r_{i,j}$, for every $j \in [N]$. For every $j \in [m]$, all the parties execute the protocol $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{\hat{p_j}\}}$. The input of P_i is $(x_i, r_{i,1},\ldots,r_{i,N})$ in this protocol. In the end, every party receives $\alpha_j = \hat{p_j}(x_1,\ldots,x_n)$, for every $j \in [m]$. Every party then executes $\mathsf{D}(\alpha_1,\ldots,\alpha_n)$ to obtain α^*. It outputs α^*.

Theorem 8. *Assuming the security of $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$ and security of RP, Π_{sh}^{F} is a robust semi-honest secure MPC protocol satisfying Definition 1 as long as $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$ satisfies Definition 1.*

The proof can be found in B.5.

4 Five Round Malicious MPC

Overview. We start by giving an overview of our construction. We want to use the robust semi honest MPC as the basis for our construction, but its security is only defined in the semi-honest setting. We enforce the semi-honest setting by having the players prove, in parallel, that they computed the robust semi honest MPC honestly. Players prove that (1) they computed the first three rounds of the robust semi honest MPC honestly; and (2) they committed their input and randomness used in the robust semi honest MPC to every other party using an extractable commitment scheme. To do so, we use a four round input delayed proof system, where the statement for the proof can be delayed till the final round. This lets players send the final round of their proof in the fourth round. Before proceeding, we verify each of the proofs received to ensure everyone is behaving in an honest manner. Next, to prove that the last round of the robust semi honest MPC is computed correctly, we use an instance of a three round input delayed witness indistinguishable proof of knowledge. The three round proof starts in the third round and completes with the last round of the robust MPC. This gives the total of five rounds.

4.1 Our Construction

Building blocks. For construction of the protocol, we require the following tools:

1. *A 3-round "rewinding-secure" extractable commitment scheme* $\Pi_{\mathsf{ecom}} = \langle C_{\mathsf{ecom}}, R_{\mathsf{ecom}} \rangle$ as described in appendix A.6. In the honest execution of our MPC protocol, we require the commitments to be *well formed*, where this property is defined in section A.6. Since there will be commitments in both directions for every pair of players, we introduce notation for individual messages of the protocol. $\pi^{\mathsf{j}}_{\mathsf{ecom}_{\mathsf{i}\to\mathsf{k}}}$ refers to the j-th round of the P_{i}'s commitment to P_{k}.

We will denote by $\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} := (\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}})$. Additionally, we shall denote by $\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ the decommitment (namely, the randomness used by the committer) of the corresponding commitment.

2. *A 4-round robust semi honest MPC protocol* Π_{rMPC} from section 3. Let $\mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}$ denote its next-message function, that for player P_i, on input $(x_\mathrm{i}, r_\mathrm{i}, \vec{m}^1, \cdots, \vec{m}^\mathrm{j})$ returns $m_i^{\mathrm{j}+1}$, the message P_i broadcasts to all other players in the $(\mathrm{j}+1)$-*th* round as a part of the protocol. Here $\vec{m}^\mathrm{j} = (m_1^\mathrm{j}, \cdots, m_n^\mathrm{j})$ consists of all the messages sent during round j of the protocol. The robust semi honest MPC also consists of a function $\mathsf{Out}^{\Pi_{\mathsf{rMPC}}}$ that computes the final output y.

3. *A 4-round input delayed parallel non-malleable zero-knowledge protocol* (refer to definition 6). We make non-black box use of the NMZK protocol of [COSV17] as described in Appendix A.5.1. We denote it as $\Pi_{\mathsf{nmzk}} = \langle P_{\mathsf{nmzk}}, V_{\mathsf{nmzk}} \rangle$.

   In our MPC construction, we use Π_{nmzk} for language L that consists of instances where, for every $\mathrm{i} \in [\mathrm{n}]$, P_i correctly computes the first 3 rounds of the robust semi honest MPC with inputs $(x_\mathrm{i}, r_\mathrm{i})$, and honestly commits to this input to every other player.

$$\begin{aligned} L = \Big\{ & (\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[n]\setminus\{\mathrm{i}\}}, \mathtt{id}_\mathrm{i}, \vec{m}_\mathrm{i} = (\vec{m}^1, \vec{m}^2, m_\mathrm{i}^3)) : \\ & \exists (x_\mathrm{i}, r_\mathrm{i}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}) \text{ s.t. } \Big((\forall\ \mathrm{k} : \tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \text{ is a } \textit{well formed} \\ & \quad \text{commitment of } ((x_\mathrm{i}, r_\mathrm{i}))\ ) \ \mathtt{AND}\ (m_\mathrm{i}^1 = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_\mathrm{i}, r_\mathrm{i}) \\ & \quad \mathtt{AND}\ m_\mathrm{i}^2 = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_\mathrm{i}, r_\mathrm{i}, \vec{m}^1)\ \mathtt{AND}\ m_\mathrm{i}^3 = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_\mathrm{i}, r_\mathrm{i}, \vec{m}^1, \vec{m}^2)\ ) \Big) \Big\} \end{aligned}$$

   Further, we define the language L_i to be the subset of L that only consists of instances where player P_i, *for a fixed* i, correctly computes the first 3 rounds of the robust semi honest MPC with inputs $(x_\mathrm{i}, r_\mathrm{i})$, and honestly commits to this input to every other player.

   In an execution of Π_{nmzk} between parties i and k where i (resp., k) plays the role of the prover (resp., verifier), we denote the message sent in the j-th round by $\pi^\mathrm{j}_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$.

   Ingredients inside [COSV17] NMZK. We recall some of the key sub-protocols used in the NMZK protocol Π_{nmzk}: (1) A "trapdoor generation" protocol that uses a standard signature scheme. We use Ver to denote the verification algorithm of the signature scheme and vk to denote a verification key. (2) A four round public-coin, extractable non-malleable commitment scheme. The transcript of an execution of the non-malleable commitment scheme is denoted as τ_{nmcom}, and the associated decommitment is denoted as $\mathsf{dec}_{\mathsf{nmcom}}$. Going further, we will refer to these sub-protocols in a non-black-box manner through the remainder of our MPC construction. Whenever necessary, we will augment the above notations with subscript $\mathrm{i} \to \mathrm{k}$ to refer to an execution between party i and k.

   For a complete formal description of Π_{nmzk}, refer to appendix A.5.1.

4. *A 3 round input delayed witness indistinguishable proof of knowledge (*WIPoK*) protocol* $\Pi_{\mathsf{WIPoK}} = (P_{\mathsf{WIPoK}}, V_{\mathsf{WIPoK}})$. We require the protocols to be public coin and instantiate them using the Lapidot-Shamir protocol [LS90].

   We use Π_{WIPoK} for a language L_{WIPoK} that consists of instances where, for every $\mathrm{i}, \mathrm{k} \in [\mathrm{n}]$, player P_i proves to player P_k that either:

   – it behaved honestly, i.e. it has a witness w such that $(x_{\widehat{L}_\mathrm{i}}, w) \in \mathsf{Rel}_{\widehat{L}_\mathrm{i}}$ (the languages $\widehat{L}$ and $\widehat{L}_\mathrm{i}$ are defined below); or

- it committed (via a *mask*) to a "trapdoor witness" $(m_1, m_2, \sigma_1, \sigma_2)$ in the non-malleable commitment $\tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$ where σ_i is a valid signature on m_i with respect to the verification key $\mathsf{vk}_{\mathrm{i}\to\mathrm{k}}$ in the trapdoor generation protocol within the NMZK.

Formally,

$$\begin{aligned}L_{\mathsf{WIPoK}} = \Big\{ &\Big(x_{\widehat{L}_\mathrm{i}}, \mathtt{id}_\mathrm{k}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, s^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{vk}_{\mathrm{i}\to\mathrm{k}} \Big) : \exists (w, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2) \text{ s.t.} \\ &\Big(\ (x_{\widehat{L}_\mathrm{i}}, w) \in \mathsf{Rel}_{\widehat{L}} \ \Big) \ \mathtt{OR} \ \Big(\big(\ ((\mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2) \oplus s^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathtt{id}_\mathrm{i}) \\ &\quad \text{is a valid decommitment of } \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}} \ ) \ \mathtt{AND} \ \mathsf{Ver}(\mathsf{vk}_{\mathrm{i}\to\mathrm{k}}, \mathsf{msg}_1, \sigma_1) = 1 \\ &\quad \mathtt{AND} \ \mathsf{Ver}(\mathsf{vk}_{\mathrm{i}\to\mathrm{k}}, \mathsf{msg}_2, \sigma_2) = 1 \ \mathtt{AND} \ \mathsf{msg}_1 \neq \mathsf{msg}_2) \big) \Big) \Big\}\end{aligned}$$

We define $x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} := \Big(x_{\widehat{L}_\mathrm{i}}, \mathtt{id}_\mathrm{k}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, s^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{vk}_{\mathrm{i}\to\mathrm{k}} \Big)$.

We now describe the language $\widehat{L}$. For every i, it consists of instances where player P_i correctly computes the fourth round of the robust semi honest MPC with inputs $(x_\mathrm{i}, r_\mathrm{i})$, and honestly commits to $(x_\mathrm{i}, r_\mathrm{i})$ to every other player P_k in the extractable commitment.

Formally,

$$\begin{aligned}\widehat{L} = \Big\{ &(\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[n]\setminus\{\mathrm{i}\}}, \mathtt{id}_\mathrm{i}, \vec{m}_\mathrm{i} = (\vec{m}^1, \vec{m}^2, \vec{m}^3, m^4_\mathrm{i})) : \\ &\exists (x_\mathrm{i}, r_\mathrm{i}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in\mathrm{n}}) \text{ s.t. } \Big((\ \forall \ \mathrm{k} : \tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \text{ is a } \textit{well formed} \\ &\quad \text{commitment of } ((x_\mathrm{i}, r_\mathrm{i}))) \ \mathtt{AND} \ (\ m^4_\mathrm{i} = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_\mathrm{i}, r_\mathrm{i}, \vec{m}^1, \vec{m}^2, \vec{m}^3) \ ) \Big) \Big\}.\end{aligned}$$

Further, we define the language $\widehat{L}_\mathrm{i}$ to be the subset of $\widehat{L}$ that only consists of instances where player P_i, *for a fixed* i, correctly computes the fourth round of the robust semi honest MPC with inputs $(x_\mathrm{i}, r_\mathrm{i})$, and honestly commits to $(x_\mathrm{i}, r_\mathrm{i})$ to every other player P_k in the extractable commitment.

Protocol description. Let $\mathcal{P} = \{P_1, \cdots, P_\mathrm{n}\}$ be the set of parties and $\{\mathtt{id}_1, \cdots, \mathtt{id}_\mathrm{n}\}$ denote their corresponding unique identifiers (one can think of $\mathtt{id}_\mathrm{i} = \mathrm{i}$). The input and randomness (x_i, r_i) for player P_i is fixed in the beginning of the protocol. The protocol instructs each player P_i to compute a message M^j_i for round j and broadcasts it over the simultaneous broadcast channel. Thus in round j, messages $(M^\mathrm{j}_1, \cdots, M^\mathrm{j}_\mathrm{n})$ are simultaneously broadcast.

The protocol is detailed below. For ease of notation, we shall assume the that security parameter n is an implicit argument to each of the functions.

Round 1. Each player P_i computes the message M^1_i to be sent in the first round as follows:

1. Compute independently, with fresh randomness, to the input and randomness in the first (committer) message of the extractable commitment to every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$(\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}) \leftarrow C_{\mathsf{ecom}}((x_\mathrm{i}, r_\mathrm{i}))$$

Set

$$\pi^1_{\mathsf{ecom}_\mathrm{i}} := (\pi^1_{\mathsf{ecom}_{\mathrm{i}\to 1}}, \cdots, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{n}}}).$$

2. Compute independently, with fresh randomness, the first (verifier) message of the non-malleable zero-knowledge protocol for every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{nmzk}}(\mathtt{id}_\mathrm{k}, \ell)$$

where ℓ is the lengths of the input delayed statements for L_{i}.

Set

$$\pi^1_{\mathsf{nmzk}_{\mathsf{i}}} := (\pi^1_{\mathsf{nmzk}_{1\to\mathsf{i}}}, \cdots, \pi^1_{\mathsf{nmzk}_{\mathsf{i}-1\to\mathsf{i}}}, \bot, \pi^1_{\mathsf{nmzk}_{\mathsf{i}+1\to\mathsf{i}}}, \cdots, \pi^1_{\mathsf{nmzk}_{\mathsf{n}\to\mathsf{i}}})$$

3. Compute the first message of the robust semi honest MPC,

$$m^1_{\mathsf{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathsf{i}}, r_{\mathsf{i}}).$$

M^1_{i} is now defined as,

$$M^1_{\mathsf{i}} := (\pi^1_{\mathsf{ecom}_{\mathsf{i}}}, \pi^1_{\mathsf{nmzk}_{\mathsf{i}}}, m^1_{\mathsf{i}}).$$

Broadcast M^1_{i} and receive $M^1_1, \cdots, M^1_{\mathsf{i}-1}, M^1_{\mathsf{i}+1}, \cdots, M^1_{\mathsf{n}}$.

Round 2. Each player P_{i} computes the message M^2_{i} to be sent in the second round as follows:

1. Compute the second message of the extractable commitment in response to the messages from the other parties i.e., $\forall \mathsf{k} \in [\mathsf{n}] \setminus \{\mathsf{i}\}$
$$\pi^2_{\mathsf{ecom}_{\mathsf{k}\to\mathsf{i}}} \leftarrow R_{\mathsf{ecom}}(\pi^1_{\mathsf{ecom}_{\mathsf{k}\to\mathsf{i}}})$$
where $\pi^1_{\mathsf{ecom}_{\mathsf{k}\to\mathsf{i}}}$ can be obtained from $\pi^1_{\mathsf{ecom}_{\mathsf{k}}}$ in M^1_{k}.

   Set

$$\pi^2_{\mathsf{ecom}_{\mathsf{i}}} := (\pi^2_{\mathsf{ecom}_{1\to\mathsf{i}}}, \cdots, \pi^2_{\mathsf{ecom}_{\mathsf{i}-1\to\mathsf{i}}}, \bot, \pi^2_{\mathsf{ecom}_{\mathsf{i}+1\to\mathsf{i}}}, \cdots, \pi^2_{\mathsf{ecom}_{\mathsf{n}\to\mathsf{i}}}).$$

2. Compute the second message of the non-malleable zero-knowledge protocol in response to the messages from the other parties i.e., $\forall \mathsf{k} \in [\mathsf{n}] \setminus \{\mathsf{i}\}$

$$\pi^2_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}} \leftarrow P_{\mathsf{nmzk}}(\mathtt{id}_{\mathsf{i}}, \ell, \pi^1_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}})$$

   where $\pi^1_{\mathsf{nmzk}_{\mathsf{k}\to\mathsf{i}}}$ can be obtained from $\pi^1_{\mathsf{nmzk}_{\mathsf{k}}}$ in M^1_{k}. Set

$$\pi^2_{\mathsf{nmzk}_{\mathsf{i}}} := (\pi^2_{\mathsf{nmzk}_{\mathsf{i}\to 1}}, \cdots, \pi^2_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{i}-1}}, \bot, \pi^2_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{i}+1}}, \cdots, \pi^2_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{n}}})$$

3. Compute the second message of the robust semi honest MPC,

$$m^2_{\mathsf{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathsf{i}}, r_{\mathsf{i}}, \vec{m}^1)$$

   where $\vec{m}^1 := (m^1_1, \cdots, m^1_{\mathsf{n}})$.

M^2_{i} is now defined as,

$$M^2_{\mathsf{i}} := (\pi^2_{\mathsf{ecom}_{\mathsf{i}}}, \pi^2_{\mathsf{nmzk}_{\mathsf{i}}}, m^2_{\mathsf{i}}).$$

Broadcast M^2_{i} and receive $M^2_1, \cdots, M^2_{\mathsf{i}-1}, M^2_{\mathsf{i}+1}, \cdots, M^2_{\mathsf{n}}$.

Round 3. Each player P_i computes the message M_i^3 to be sent in the third round as follows:

1. Compute the final message of the extractable commitment i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \leftarrow C_{\mathsf{ecom}}(\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}})$$

   where $\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ is as computed earlier and $\pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^2_{\mathsf{ecom}_{\mathrm{k}}}$ in M^2_k. Set

$$\pi^3_{\mathsf{ecom}_\mathrm{i}} := (\pi^3_{\mathsf{ecom}_{\mathrm{i}\to 1}}, \cdots, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{n}}}).$$

2. Compute the third message of the non-malleable zero-knowledge protocols i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^3_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{nmzk}}(\mathtt{id}_\mathrm{k}, \pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}})$$

   where $\pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$ is as computed earlier and $\pi^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$ is obtained from $\pi^2_{\mathsf{nmzk}_\mathrm{k}}$ in M^2_k.
   Set

$$\pi^3_{\mathsf{nmzk}_\mathrm{i}} := (\pi^3_{\mathsf{nmzk}_{1\to\mathrm{i}}}, \cdots, \pi^3_{\mathsf{nmzk}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^3_{\mathsf{nmzk}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^3_{\mathsf{nmzk}_{\mathrm{n}\to\mathrm{i}}})$$

3. Compute the third message of the robust semi honest MPC,

$$m^3_\mathrm{i} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_\mathrm{i}, r_\mathrm{i}, \vec{m}^1, \vec{m}^2)$$

   where $\vec{m}^1 := (m^1_1, \cdots, m^1_\mathrm{n})$ and $\vec{m}^2 := (m^2_1, \cdots, m^2_\mathrm{n})$.

4. Compute the first message for the input delayed witness indistinguishable proof of knowledge (WIPoK) for $\widehat{L}_{\mathsf{WIPoK}}$ to every other player i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} \leftarrow P_{\mathsf{WIPoK}}(\widehat{\ell})$$

   where $\widehat{\ell}$ is the size of the statement.

   Set

$$\pi^1_{\mathsf{WIPoK}_\mathrm{i}} := (\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to 1}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{n}}}).$$

M^3_i is now defined as,

$$M^3_\mathrm{i} := (\pi^3_{\mathsf{ecom}_\mathrm{i}}, \pi^3_{\mathsf{nmzk}_\mathrm{i}}, m^3_\mathrm{i}, \pi^1_{\mathsf{WIPoK}_\mathrm{i}}).$$

Broadcast M^3_i and receive $M^3_1, \cdots, M^3_{\mathrm{i}-1}, M^3_{\mathrm{i}+1}, \cdots, M^3_\mathrm{n}$.

Round 4. Each player P_i computes the message M^4_i to be sent in the fourth round as follows:

1. Compute the second message of the input delayed WIPoK for L_{WIPoK} in response to messages from every other player i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^2_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{WIPoK}}(\ell, \pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}})$$

   where $\pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}}$ can be obtained from $\pi^1_{\mathsf{WIPoK}_\mathrm{k}}$ in M^3_k.

   Set

$$\pi^1_{\mathsf{WIPoK}_\mathrm{i}} := (\pi^1_{\mathsf{WIPoK}_{1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{n}\to\mathrm{i}}}).$$

2. Compute the final message of the non-malleable zero-knowledge protocol for language L_{i} i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\begin{aligned} w_{L_{\mathrm{i}}} &\coloneqq \left(x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}\right) \\ \vec{m}_{\mathrm{i}} &\coloneqq \left(\vec{m}^1, \vec{m}^2, m^3_{\mathrm{i}}\right) \\ x_{L_{\mathrm{i}}} &\coloneqq \left(\left\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{k}\in[\mathrm{n}]}, \mathtt{id}_{\mathrm{i}}, \vec{m}_{\mathrm{i}}\right) \\ \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}} &\leftarrow P_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{i}}, \ell, x_{L_{\mathrm{i}}}, w_{L_{\mathrm{i}}}, \pi^1_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}, \pi^3_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}) \end{aligned}$$

where $|x_{L_{\mathrm{i}}}| = \ell$, and $\pi^1_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^1_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^1_{k}. Similarly, $\pi^3_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^3_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^3_{k}. $\pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is as computed earlier.

Set

$$\pi^4_{\mathsf{nmzk}_{\mathrm{i}}} \coloneqq (\pi^4_{\mathsf{nmzk}_{\mathrm{i}\to 1}}, \cdots, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}-1}}, \bot, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{n}}}).$$

M^4_{i} is now defined as,

$$M^4_{\mathrm{i}} \coloneqq (\pi^4_{\mathsf{nmzk}_{\mathrm{i}}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{i}}}).$$

Broadcast M^4_{i} and receive $M^4_1, \cdots, M^4_{\mathrm{i}-1}, M^4_{\mathrm{i}+1}, \cdots, M^4_{\mathrm{n}}$.

Round 5. Each player P_{i} computes the message M^5_{i} to be sent in the fifth round as follows:

1. Check if all the proofs for the NMZK in the protocol are accepting. The proof from P_{k} to P_{j} is accepting if P_{k} has computed the first 3 rounds of the robust semi honest MPC correctly and has committed to the same inputs, used in the robust semi honest MPC, to every other player.

   First, compute the statement $x_{\mathsf{nmzk}_{\mathrm{k}}}$ for each player P_{k} i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\begin{aligned} \vec{m}_{\mathrm{k}} &\coloneqq \left(\vec{m}^1, \vec{m}^2, m^3_{\mathrm{k}}\right) \\ x_{L_{\mathrm{k}}} &\coloneqq \left(\left\{\tau_{\mathsf{ecom}_{\mathrm{k}\to\mathrm{t}}}\right\}_{\mathrm{t}\in[\mathrm{n}]}, \mathtt{id}_{\mathrm{k}}, \vec{m}_{\mathrm{k}}\right) \end{aligned}$$

   Next, check if every proof is valid.

$$\begin{aligned} &\text{if } \exists \mathrm{k}, \mathrm{j} \text{ s.t } \mathsf{accept} \neq V_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{k}}, x_{L_{\mathrm{k}}}, \pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}, \pi^3_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}, \pi^4_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}) \\ &\qquad\qquad \text{then } \mathsf{output} \perp \text{ and } \mathsf{abort} \\ &\quad \text{else continue} \end{aligned}$$

   This can be done because the proofs are public coin. Moreover this is done to avoid the case that some honest parties continue on to the next round, but the others abort.

2. Compute the final message of the robust semi honest MPC,

$$m^4_{\mathrm{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2, \vec{m}^3)$$

   where $\vec{m}^1 \coloneqq (m^1_1, \cdots, m^1_{\mathrm{n}})$, $\vec{m}^2 \coloneqq (m^2_1, \cdots, m^2_{\mathrm{n}})$ and $\vec{m}^3 \coloneqq (m^3_1, \cdots, m^3_{\mathrm{n}})$.

3. Compute the final message of WIPoK for language L_{WIPoK} i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\begin{aligned} \widehat{w}_{\widehat{L}_{\mathrm{i}}} &\coloneqq \left(x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}\right) \\ \vec{m}_{\mathrm{i}} &\coloneqq \left(\vec{m}^1, \vec{m}^2, \vec{m}^3, m^4_{\mathrm{i}}\right) \\ \widehat{x}_{\widehat{L}_{\mathrm{i}}} &\coloneqq \left(\left\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{k}\in[\mathrm{n}]}, \mathtt{id}_{\mathrm{i}}, \vec{m}_{\mathrm{i}}\right) \\ x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} &\coloneqq \left(x_{\widehat{L}_{\mathrm{i}}}, \mathtt{id}_{\mathrm{k}}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, s^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{vk}_{\mathrm{i}\to\mathrm{k}}\right) \\ w_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} &\coloneqq (\widehat{w}_{\widehat{L}_{\mathrm{i}}}, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \rho) \\ \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} &\leftarrow P_{\mathsf{WIPoK}}(\mathtt{id}_{\mathrm{i}}, \widehat{\ell}, x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}, \widehat{w}_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}) \end{aligned}$$

where $|x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}| = \widehat{\ell}$, and $\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$ is as computed earlier and $\pi^2_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^2_{\mathsf{WIPoK}_{\mathrm{k}}}$ in M^4_{k}.

Set

$$\pi^3_{\mathsf{WIPoK}_{\mathrm{i}}} := (\pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to 1}}, \cdots, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to \mathrm{i}-1}}, \bot, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to \mathrm{i}+1}}, \cdots, \widehat{\pi}^3_{\mathsf{WIPoK}_{\mathrm{i}\to \mathrm{n}}})$$

M^5_{i} is now defined as,

$$M^5_{\mathrm{i}} := (m^4_{\mathrm{i}}, \widehat{\pi}^3_{\mathsf{WIPoK}_{\mathrm{i}}}).$$

Broadcast M^5_{i} and receive $M^5_1, \cdots, M^5_{\mathrm{i}-1}, M^5_{\mathrm{i}+1}, \cdots, M^5_{\mathrm{n}}$.

Output computation. To compute the output, P_{i} performs the following steps:

1. Check if all the proofs in the protocol for the WIPoK are accepting. The proof from P_{k} to P_{j} is accepting if P_{k} has computed the 4-th round of the robust semi honest MPC correctly and has committed to the same inputs, used in the robust semi honest MPC, to every other party.

   Check if every proof is valid.

   $$\text{if } \exists \mathrm{k}, \mathrm{j} \text{ s.t } \mathsf{accept} \neq V_{\mathsf{WIPoK}}(\mathtt{id}_{\mathrm{k}}, x_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^3_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}})$$
   $$\text{then } \mathsf{output} \perp \text{ and } \mathsf{abort}$$
   $$\text{else continue}$$

2. Compute the output of the protocol as

   $$y \leftarrow \mathsf{Out}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2, \vec{m}^3, \vec{m}^4)$$

Theorem 9. *Assuming security of the building blocks, the above described five round protocol is secure against malicious adversaries.*

Extractable commitments, WIPoK and NMZK can be instantiated from DL, while the robust semi-honest MPC can be instantiated from DDH. Thus, all the required primitives can be instantiated from DDH. The proof of theorem 9 can be found in appendix C.

5 Four Round Malicious MPC

Overview. We give an overview of our four round construction. At a high-level, the four round protocol is very similar to the five round protocol (from the previous section) but to compress the number of rounds, we construct (using sub-exponential assumptions) and use a 3 round input delayed strong WI argument of knowledge (with appropriate non-malleability properties), ending in the third round, to enable parties to prove their honest behavior of the first three rounds. This lets the players send the fourth message in the clear if the proof at the end of the third round verifies. For the output round, we use a four-round NMZK as before to prove honest behavior.

We now describe our three-round input delayed proof system. We build it in the simultaneous-message model, where the prover and the verifier send messages simultaneously in the first two rounds. (However, security holds even against rushing adversaries). To prove an instance x in a language L, the protocol proceeds as follows:

- The verifier sends an image $y = f(r)$ of a one way permutation f on a random string r.

- The prover committs to 0 using a 2-round non-malleable commitment [KS17].

– The prover additionally gives a three-round input delayed witness indistinguishable proof of knowledge (WIPoK) proving knowledge of either:

   1. w such that $(x, w) \in \mathsf{Rel}_L$; or
   2. the decommitment of the non-malleable commitment to r' such that r' is the pre-image of y w.r.t. f.

The protocol, in the simultaneous-message model, is described in figure 1. We do not argue its security separately, but within the security proof of our MPC protocol. We remark that while the above construction uses a 2-round extractable non-malleable commitment scheme, we can also use a 3-round public-coin, extractable non-malleable commitments that are input delayed [COSV16, GPR16]. (See Appendix A.4 for further discussion.)

Informally speaking, one can think of the above construction as a strong input delayed WI argument of knowledge with non-malleability properties.

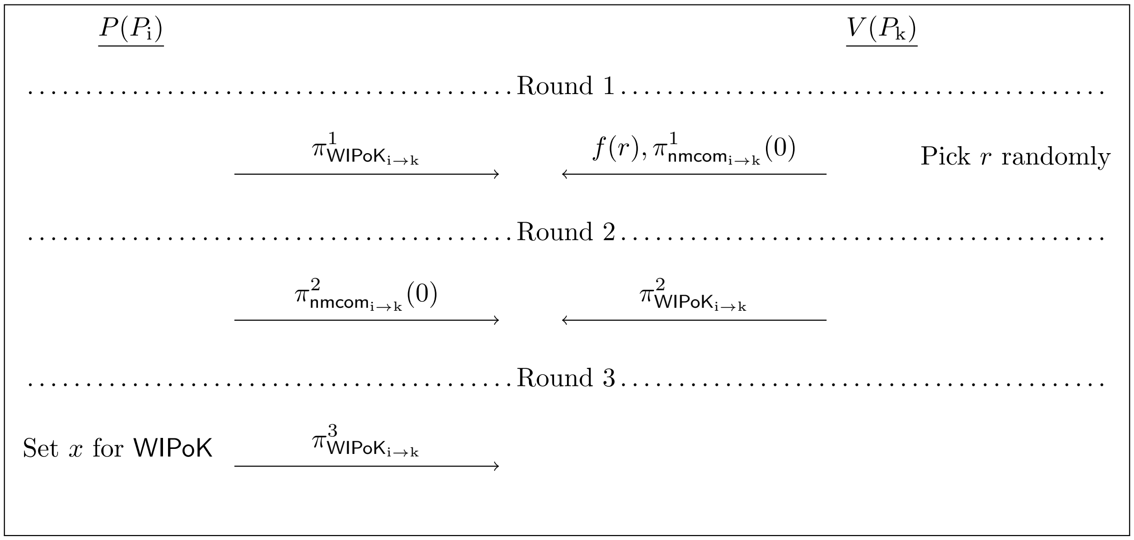
$P(P_{\mathrm{i}})$
$V(P_{\mathrm{k}})$
Round 1
$\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$
$f(r), \pi^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}(0)$
Pick r randomly
Round 2
$\pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}(0)$
$\pi^2_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$
Round 3
Set x for WIPoK
$\pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$

Figure 1: Three-round input delayed proof system

5.1 Our Construction

Building Blocks. We describe the main tools used in our MPC construction. The exact security levels for each of these primitives are discussed at the end of the section.

1. *A 3-round "rewinding secure" extractable commitment scheme* $\Pi_{\mathsf{ecom}} = \langle C_{\mathsf{ecom}}, R_{\mathsf{ecom}} \rangle$ as described in appendix A.6. In the honest execution of the protocol we require the commitments to be *well formed*, where this property is defined in section A.6. We use the notation $\pi^{\mathrm{j}}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ to denote the j-th round of an execution of Π_{ecom} where party P_{i} is the committer and party P_{k} is the receiver. The entire transcript of the execution is denoted as $\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} := \left(\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\right)$. The associated decommitment is denoted as $\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$.

   The above commitment scheme was also used in the construction of our five round MPC protocol described in Section 4. Here, we actually require an *input delayed* extractable commitment scheme. Such a commitment scheme can be obtained by a simple augmentation to protocol $\Pi_{\mathsf{ecom}} = \langle C_{\mathsf{ecom}}, R_{\mathsf{ecom}} \rangle$. To commit to a message m, we first commit to a random string r^0_{ecom} in $\Pi_{\mathsf{ecom}} = \langle C_{\mathsf{ecom}}, R_{\mathsf{ecom}} \rangle$. In the third round, in addition to the protocol message of

$\Pi_{\mathsf{ecom}} = \langle C_{\mathsf{ecom}}, R_{\mathsf{ecom}} \rangle$, we send $r^1_{\mathsf{ecom}} := r^0_{\mathsf{ecom}} \oplus m$. In the sequel, we refer to the string r^1_{ecom} as the *mask*.

2. A *one-way permutation* f.

3. An instance of *a 2-round extractable non-malleable commitment scheme* $\Pi_{\mathsf{nmcom}} = \langle C_{\mathsf{nmcom}}, R_{\mathsf{nmcom}} \rangle$. Importantly, we require that the scheme is extractable by rewinding, and both hiding and non-malleability of the non-malleable commitment hold against adversaries running in time T, where T is the running time of the rewinding extractor. This requirement is described formally at the end of the protocol. The commitment scheme of [KS17] based on sub-exponential DDH satisfies these properties.

   We use the notation $\pi^{j}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$ to denote the j-th round of an execution of Π_{nmcom} where party P_{i} is the committer and party P_{k} is the receiver. The entire transcript of the execution is denoted as $\tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}} := \big(\pi^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}\big)$. The associated decommitment is denoted as $\mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$.

4. *A 4-round robust semi honest MPC protocol* Π_{rMPC} from section 3. Let $\mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}$ denote its next-message function, that for player P_{i}, on input $(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \cdots, \vec{m}^{\mathrm{j}})$ returns $m_i^{\mathrm{j}+1}$, the message P_{i} broadcasts to all other players in the $(\mathrm{j}+1)$-*th* round as a part of the protocol. Here $\vec{m}^{\mathrm{j}} = (m^{\mathrm{j}}_1, \cdots, m^{\mathrm{j}}_n)$ consists of all the messages sent during round j of the protocol. The robust semi honest MPC also consists of a function $\mathsf{Out}^{\Pi_{\mathsf{rMPC}}}$ that computes the final output y.

5. A 3 round input delayed witness indistinguishable proof of knowledge (WIPoK) protocol $\Pi_{\mathsf{WIPoK}} = (P_{\mathsf{WIPoK}}, V_{\mathsf{WIPoK}})$. We require the protocol to be public coin and instantiate it using the Lapidot-Shamir protocol [LS90].

   We use Π_{WIPoK} for a language L_{WIPoK} that consists of instances where, for every $\mathrm{i}, \mathrm{k} \in [\mathrm{n}]$, player P_{i} proves to player P_{k} that either

   – it behaved honestly, i.e. it has a witness w such that $(x_{L_{\mathrm{i}}}, w) \in \mathsf{Rel}_{L_{\mathrm{i}}}$ (where the languages L and L_{i} are defined below); or

   – it commits to a value ρ in the non-malleable commitment such that ρ is the pre-image of a random image y of the one-way permutation f that was sent by the verifier.

Formally,

$$
\begin{aligned}
L_{\mathsf{WIPoK}} = \Big\{ & \big(x_{L_{\mathrm{i}}}, \mathtt{id}_{\mathrm{k}}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, y_{\mathrm{k}\to\mathrm{i}}\big) : \exists (w, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \rho) \text{ s.t.} \\
& \Big(\ (x_{L_{\mathrm{i}}}, w) \in \mathsf{Rel}_L \ \Big) \ \mathtt{OR} \ \Big(f(\rho) = y_{\mathrm{k}\to\mathrm{i}} \ \mathtt{AND} \ (\ \ (\rho, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathtt{id}_{\mathrm{i}}) \\
& \quad \text{is a valid decommitment of } \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}} \ \ )\Big)\Big\}
\end{aligned}
$$

We define $x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} := (x_{L_{\mathrm{i}}}, \mathtt{id}_{\mathrm{k}}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, y_{\mathrm{k}\to\mathrm{i}})$.

We now define the languages L and L_i. Informally, L is the language which consists of instances where, for every $i \in [\mathrm{n}]$, player P_{i} correctly computes the first three rounds of the robust semi honest MPC with inputs $(x_{\mathrm{i}}, r_{\mathrm{i}})$, and honestly commits to $(x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ to every other player P_{k} in the extractable commitment.

$$
\begin{aligned}
L = \Big\{ & (\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[n]\setminus\{\mathrm{i}\}}, \mathtt{id}_{\mathrm{i}}, \vec{m}_{\mathrm{i}} = (\vec{m}^1, \vec{m}^2, m^3_{\mathrm{i}})) : \\
& \exists (x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}) \text{ s.t. } \Big((\forall \ \mathrm{k} : \tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \text{ is a } \textit{well formed} \text{ commitment of } \big((x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\big)\big) \\
& \quad \mathtt{AND} \ (m^1_{\mathrm{i}} = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}) \ \mathtt{AND} \\
& \quad m^2_{\mathrm{i}} = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1) \ \mathtt{AND} \ m^3_{\mathrm{i}} = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2) \ )\Big)\Big\}
\end{aligned}
$$

Further, we define the language L_{i} to be the subset of L that only consists of instances where player P_{i}, *for a fixed* i, correctly computes the first three rounds of the robust semi honest MPC with inputs $(x_{\mathrm{i}}, r_{\mathrm{i}})$, and honestly commits to $(x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ to every other player P_{k} in the extractable commitment.

6. *A 4-round input delayed parallel non-malleable zero-knowledge protocols* (refer to definition 6). We will use the protocol of [COSV17] in a non black-box manner as described in appendix A.5.1. We denote it as $\Pi_{\mathsf{nmzk}} = \langle P_{\mathsf{nmzk}}, V_{\mathsf{nmzk}} \rangle$.

   In our MPC construction, we use Π_{nmzk} for the language $\widehat{L}$. Informally, $\widehat{L}$ is the language which consists of instances where, for every $\mathrm{i} \in [\mathrm{n}]$, player P_{i} (a) correctly computed the final round of the robust MPC with inputs $(x_{\mathrm{i}}, r_{\mathrm{i}})$; and (b) commits to $(x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ to every other player P_{k} in the extractable commitment such that they are well formed.

   Formally,

$$\begin{aligned}\widehat{L} = \Big\{ &(\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[n]\setminus\{\mathrm{i}\}}, \mathtt{id}_{\mathrm{i}}, \vec{m}_{\mathrm{i}} = (\vec{m}^1, \vec{m}^2, \vec{m}^3, m^4_{\mathrm{i}})) : \\ &\quad \exists (x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in\mathrm{n}}) \text{ s.t. } \Big((\ \forall \ \mathrm{k} : \tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \text{ is a } \textit{well formed} \\ &\qquad \text{commitment of } \big((x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\big) \ \mathtt{AND} \ (\ m^4_{\mathrm{i}} = \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2, \vec{m}^3) \ ) \Big) \Big\}.\end{aligned}$$

   Further, we define the language $\widehat{L}_{\mathrm{i}}$ to be the subset of $\widehat{L}$ that only consists of instances where player P_{i}, *for a fixed* i, (a) correctly computed the final round of the robust MPC with inputs $(x_{\mathrm{i}}, r_{\mathrm{i}})$; and (b) commits to $(x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ to every other player P_{k} in the extractable commitment such that they are well formed.

   In an execution of Π_{nmzk} between parties i and k where i (resp., k) plays the role of the prover (resp., verifier), we denote the message sent in the j-th round by $\pi^{\mathrm{j}}_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$.

Protocol description.

Round 1. Each player P_{i} computes the message M^1_{i} to be sent in the first round as follows:

1. Compute independently, with fresh randomness, the first (committer) message of the extractable commitment for every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$r^0_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \xleftarrow{\$} \{0,1\}^{|(x_i, r_i)|}$$
$$(\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{dec}_{\mathrm{i}\to\mathrm{k}}) \leftarrow C_{\mathsf{ecom}}(r^0_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}})$$

   Set

$$\pi^1_{\mathsf{ecom}_{\mathrm{i}}} := (\pi^1_{\mathsf{ecom}_{\mathrm{i}\to1}}, \cdots, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}-1}}, \bot, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{n}}}).$$

2. Compute the first message of the robust semi honest MPC,

$$m^1_{\mathrm{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}})$$

3. Compute the different components that will make up the proof system for L.

   (a) Select random strings independently and compute its image with respect to the function f. that will serve as the basis for the trapdoor, and apply the function f to send to every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\rho_{\mathrm{i}\to\mathrm{k}} \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(n)}$$
$$y_{\mathrm{i}\to\mathrm{k}} := f(\rho_{\mathrm{i}\to\mathrm{k}})$$

Set

$$y_{\mathrm{i}} := (y_{\mathrm{i}\to 1}, \cdots, y_{\mathrm{i}\to \mathrm{i}-1}, \perp, y_{\mathrm{i}\to \mathrm{i}+1}, \cdots, y_{\mathrm{i}\to \mathrm{n}}).$$

Looking ahead, the $\rho_{\mathrm{i}\to\mathrm{k}}$ will be used as the basis for the "trapdoor" witness.

(b) Send the first (receiver) message of the non-malleable commitment to every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^1_{\mathsf{nmcom}_{\mathrm{k}\to\mathrm{i}}} \leftarrow R_{\mathsf{nmcom}}(1^n)$$

Set

$$\pi^1_{\mathsf{nmcom}_{\mathrm{i}}} := (\pi^1_{\mathsf{nmcom}_{1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{nmcom}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^1_{\mathsf{nmcom}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{nmcom}_{\mathrm{n}\to\mathrm{i}}}).$$

(c) Compute the first message for the input delayed witness indistinguishable proof of knowledge (WIPoK) for L_{WIPoK} to every other player i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} \leftarrow P_{\mathsf{WIPoK}}(\ell)$$

where ℓ is the size of the statement.

Set

$$\pi^1_{\mathsf{WIPoK}_{\mathrm{i}}} := (\pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to 1}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{n}}}).$$

4. Compute independently, with fresh randomness, the first (verifier) message of the non-malleable zero-knowledge protocol for every other player i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{k}}, \widehat{\ell})$$

where $\widehat{\ell}$ is the length of the input delayed statement for $\widehat{L}$.

Set

$$\pi^1_{\mathsf{nmzk}_{\mathrm{i}}} := (\pi^1_{\mathsf{nmzk}_{1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{nmzk}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^1_{\mathsf{nmzk}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{nmzk}_{\mathrm{n}\to\mathrm{i}}})$$

M^1_{i} is now defined as,

$$M^1_{\mathrm{i}} := (\pi^1_{\mathsf{ecom}_{\mathrm{i}}}, y_i, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}}}, \pi^1_{\mathsf{nmcom}_{\mathrm{i}}}, \pi^1_{\mathsf{nmzk}_{\mathrm{i}}}, m^1_{\mathrm{i}})$$

Broadcast M^1_{i} and receive $M^1_1, \cdots, M^1_{\mathrm{i}-1}, M^1_{\mathrm{i}+1}, \cdots, M^1_{\mathrm{n}}$.

Round 2. Each player P_{i} computes the message M^2_{i} to be sent in the second round as follows:

1. Compute the second message of the extractable commitment in response to the messages from the other parties i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^2_{\mathsf{ecom}_{\mathrm{k}\to\mathrm{i}}} \leftarrow R_{\mathsf{ecom}}(\pi^1_{\mathsf{ecom}_{\mathrm{k}\to\mathrm{i}}})$$

where $\pi^1_{\mathsf{ecom}_{\mathrm{k}\to\mathrm{i}}}$ can be obtained from $\pi^1_{\mathsf{ecom}_{\mathrm{k}}}$ in M^1_{k}.

Set

$$\pi^2_{\mathsf{ecom}_{\mathrm{i}}} := (\pi^2_{\mathsf{ecom}_{1\to\mathrm{i}}}, \cdots, \pi^2_{\mathsf{ecom}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^2_{\mathsf{ecom}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^2_{\mathsf{ecom}_{\mathrm{n}\to\mathrm{i}}}).$$

2. Compute the second message of the robust semi honest MPC,

$$m^2_{\mathrm{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1)$$

where $\vec{m}^1 := (m^1_1, \cdots, m^1_{\mathrm{n}})$.

3. Compute the second message for the different components in the proof system for L.

(a) Compute the second (final) message of the non-malleable commitment scheme in response to the messages from the other parties by committing to the '0' string i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$(\pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, \mathsf{dec}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}) \leftarrow C_{\mathsf{nmcom}}(0, \pi^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}})$$

where $\pi^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$ can be obtained from $\pi^1_{\mathsf{nmcom}_{\mathrm{k}}}$ in M^1_{k}.
Set

$$\pi^2_{\mathsf{nmcom}_{\mathrm{i}}} := (\pi^2_{\mathsf{nmcom}_{\mathrm{i}\to 1}}, \cdots, \pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^2_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{n}}}).$$

(b) Compute the second message of the input delayed WIPoK for L_{WIPoK} in response to messages from every other player i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^2_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{WIPoK}}(\ell, \pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}})$$

where $\pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}}$ can be obtained from $\pi^1_{\mathsf{WIPoK}_{\mathrm{k}}}$ in M^1_{k}.
Set

$$\pi^1_{\mathsf{WIPoK}_{\mathrm{i}}} := (\pi^1_{\mathsf{WIPoK}_{1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^1_{\mathsf{WIPoK}_{\mathrm{n}\to\mathrm{i}}}).$$

4. Compute the second message of the non-malleable zero-knowledge protocols in response to the messages from the other parties i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}} \leftarrow P_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{i}}, \widehat{\ell}, \pi^1_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}})$$

where $\pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$ can be obtained from $\pi^1_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^1_{k}. Set

$$\pi^2_{\mathsf{nmzk}_{\mathrm{i}}} := (\pi^2_{\mathsf{nmzk}_{\mathrm{i}\to 1}}, \cdots, \pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{n}}})$$

M^2_{i} is now defined as,

$$M^2_{\mathrm{i}} := (\pi^2_{\mathsf{ecom}_{\mathrm{i}}}, \pi^2_{\mathsf{nmcom}_{\mathrm{i}}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{i}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{i}}}, m^2_{\mathrm{i}}).$$

Broadcast M^2_{i} and receive $M^2_1, \cdots, M^2_{\mathrm{i}-1}, M^2_{\mathrm{i}+1}, \cdots, M^2_{\mathrm{n}}$.

Round 3. Each player P_{i} computes the message M^1_{i} to be sent in the third round as follows:

1. Compute the final message of the extractable commitment i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \leftarrow C_{\mathsf{ecom}}(\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}})$$

where $\pi^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ is as computed earlier and $\pi^2_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^2_{\mathsf{ecom}_{\mathrm{k}}}$ in M^2_{k}.
Set $\pi^3_{\mathsf{ecom}_{\mathrm{i}}} := (\pi^3_{\mathsf{ecom}_{\mathrm{i}\to 1}}, \cdots, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^3_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{n}}})$.

2. Compute $(x_{\mathrm{i}}, r_{\mathrm{i}})$ masked with the random string sent in the extractable commitment, i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} := r^0_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \oplus (x_{\mathrm{i}}, r_{\mathrm{i}})$$

Set Set $r^1_{\mathsf{ecom}_{\mathrm{i}}} := (r^1_{\mathsf{ecom}_{\mathrm{i}\to 1}}, \cdots, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{n}}})$.

3. Compute the third message of the robust semi honest MPC,

$$m^3_{\mathrm{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2)$$

where $\vec{m}^1 := (m^1_1, \cdots, m^1_{\mathrm{n}})$ and $\vec{m}^2 := (m^2_1, \cdots, m^2_{\mathrm{n}})$.

4. Compute the third message for the different components in the proof system for L.

(a) Set the statement and witness for the input delayed WIPoK language L_{WIPoK}.

$$\begin{aligned}
\vec{m} &:= \left(\vec{m}^1, \vec{m}^2, m_{\mathrm{i}}^3\right) \\
x_{L_{\mathrm{i}}} &:= \left(\left\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{k}\in[\mathrm{n}]}, \mathtt{id}_{\mathrm{i}}, \vec{m}\right) \\
w_{L_{\mathrm{i}}} &:= \left(x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}\right) \\
\forall \mathrm{k} : x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} &:= \left(x_{L_{\mathrm{i}}}, \mathtt{id}_{\mathrm{k}}, \tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}, y_{\mathrm{k}\to\mathrm{i}}, s^1_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}\right) \\
\forall \mathrm{k} : w_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} &:= \left(w_{L_{\mathrm{i}}}, \perp, \perp\right)
\end{aligned}$$

where $\forall \mathrm{k} : |x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}| = \ell$.
Compute the final message of the WIPoK for language L_{WIPoK}, i.e. $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}} \leftarrow P_{\mathsf{WIPoK}}(x_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}, w_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}, \pi^1_{\mathsf{WIPoK}_{\mathrm{i}}\to\mathrm{k}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{i}}\to\mathrm{k}})$$

Set $\pi^3_{\mathsf{WIPoK}_{\mathrm{i}}} := (\pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to1}}, \cdots, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}-1}}, \perp, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{n}}})$.

5. Compute the third message of the non-malleable zero-knowledge protocol i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\pi^3_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}} \leftarrow V_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{k}}, \pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}})$$

where $\pi^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$ is as computed earlier and $\pi^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$ is obtained from $\pi^2_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^2_{k}.
Set

$$\pi^3_{\mathsf{nmzk}_{\mathrm{i}}} := (\pi^3_{\mathsf{nmzk}_{1\to\mathrm{i}}}, \cdots, \pi^3_{\mathsf{nmzk}_{\mathrm{i}-1\to\mathrm{i}}}, \perp, \pi^3_{\mathsf{nmzk}_{\mathrm{i}+1\to\mathrm{i}}}, \cdots, \pi^3_{\mathsf{nmzk}_{\mathrm{n}\to\mathrm{i}}})$$

M^3_{i} is now defined as,

$$M^3_{\mathrm{i}} := (\pi^3_{\mathsf{ecom}_{\mathrm{i}}}, r^1_{\mathsf{ecom}_{\mathrm{i}}}, \pi^3_{\mathsf{WIPoK}_{\mathrm{i}}}, \pi^3_{\mathsf{nmzk}_{\mathrm{i}}}, m^3_{\mathrm{i}}).$$

Broadcast M^3_{i} and receive $M^3_1, \cdots, M^3_{\mathrm{i}-1}, M^3_{\mathrm{i}+1}, \cdots, M^3_{\mathrm{n}}$.

Round 4. Each player P_{i} computes the message M^1_{i} to be sent in the fourth round as follows:

1. Check if all the proofs in the protocol L_{WIPoK} are accepting. Compute, as earlier, the statement $x_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}$ for every player P_{k} and P_{j}.
Next, check if every proof is valid.

$$\begin{aligned}
&\text{if } \exists \mathrm{k}, \mathrm{j} \text{ s.t } \mathsf{accept} \neq V_{\mathsf{WIPoK}}(x_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^1_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^2_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}, \pi^3_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{j}}}) \\
&\qquad\qquad \text{then } \mathsf{output} \perp \text{ and } \mathsf{abort} \\
&\qquad \text{else continue}
\end{aligned}$$

2. Compute the final message of the robust semi honest MPC,

$$m^4_{\mathrm{i}} \leftarrow \mathsf{nextMsg}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2, \vec{m}^3)$$

where $\vec{m}^1 := (m^1_1, \cdots, m^1_{\mathrm{n}})$, $\vec{m}^2 := (m^2_1, \cdots, m^2_{\mathrm{n}})$ and $\vec{m}^3 := (m^3_1, \cdots, m^3_{\mathrm{n}})$.

3. Compute the final message of the non-malleable zero-knowledge protocol for language $\widehat{L}_{\mathrm{i}}$ i.e., $\forall \mathrm{k} \in [\mathrm{n}] \setminus \{\mathrm{i}\}$

$$\begin{aligned}
w_{\widehat{L}_{\mathrm{i}}} &:= \left(x_{\mathrm{i}}, r_{\mathrm{i}}, \{\mathsf{dec}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{\mathrm{k}\in[\mathrm{n}]}\right) \\
\vec{m}_{\mathrm{i}} &:= \left(\vec{m}^1, \vec{m}^2, \vec{m}^3, m^4_{\mathrm{i}}\right) \\
x_{\widehat{L}_{\mathrm{i}}} &:= \left(\left\{\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}, r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{k}\in[\mathrm{n}]}, \mathtt{id}_{\mathrm{i}}, \vec{m}_{\mathrm{i}}, \right)
\end{aligned}$$

$$\pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}} \leftarrow P_{\mathsf{nmzk}}(\mathtt{id}_{\mathrm{i}}, \widehat{\ell}, x_{\widehat{L}_{\mathrm{i}}}, w_{\widehat{L}_{\mathrm{i}}}, \pi^1_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}, \pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}, \pi^3_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}})$$

where $\pi^1_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is obtained from $\pi^1_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^1_{k}. Similarly, $\pi^3_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is be obtained from $\pi^3_{\mathsf{nmzk}_{\mathrm{k}}}$ in M^3_{k}. $\pi^2_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}$ is as computed earlier.

Set

$$\pi^4_{\mathsf{nmzk}_{\mathrm{i}}} := (\pi^4_{\mathsf{nmzk}_{\mathrm{i}\to 1}}, \cdots, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}-1}}, \bot, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{i}+1}}, \cdots, \pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{n}}})$$

M^4_{i} is now defined as, $M^4_{\mathrm{i}} := (m^4_{\mathrm{i}}, \pi^4_{\mathsf{nmzk}_{\mathrm{i}}})$. Broadcast M^4_{i} and receive $M^4_1, \cdots, M^4_{\mathrm{i}-1}, M^4_{\mathrm{i}+1}, \cdots, M^4_{\mathrm{n}}$.

Output Computation. To compute the output, P_{i} performs the following steps:

1. Check if all the proofs in the protocol for $\widehat{L}$ are accepting. As before, compute the statement $x_{\widehat{L}_{\mathrm{k}}}$ for each player P_{k}.

   Next, check if every proof is valid.

   $$\text{if } \exists \mathrm{k}, \mathrm{j} \text{ s.t } \mathsf{accept} \neq \widehat{V}_{\mathsf{nmzk}}(x_{\widehat{L}_{\mathrm{k}}}, \widehat{\pi}^1_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}, \widehat{\pi}^2_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}}, \widehat{\pi}^3_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{j}}})$$
   $$\text{then } \mathsf{output} \perp \text{ and } \mathsf{abort}$$
   $$\text{else continue}$$

2. Compute the output of the protocol as

   $$y \leftarrow \mathsf{Out}^{\Pi_{\mathsf{rMPC}}}(x_{\mathrm{i}}, r_{\mathrm{i}}, \vec{m}^1, \vec{m}^2, \vec{m}^3, \vec{m}^4)$$

This completes the description of the protocol.

Leveled security. We assume the following, and set the security parameters for the primitives accordingly.

- $T_f >> \widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$;
- $T^{\mathsf{h}}_{\mathsf{nmcom}}, T^{\mathsf{nm}}_{\mathsf{nmcom}} >> T_f$;
- $T_{\mathsf{WIPoK}} >> T_f, T_{\mathsf{Sign}}, T_{\mathsf{ecom}}$;
- $T_{\mathsf{rMPC}_{(1-3)}} >> T_f, T_{\mathsf{Sign}}, T_{\mathsf{ecom}}$;
- $T_{\mathsf{ecom}} >> T_f$.

where T_{prim} means that the primitive prim is secure against adversaries running in time T_{prim}, and $T' >> T$ means that $T' > T \cdot \mathsf{poly}(n)$. Specifically $T_{\mathsf{rMPC}_{(1-3)}}$ means that we require the first three rounds of our robust MPC to be indistinguishable (for adversaries running in time $T_{\mathsf{rMPC}_{(1-3)}}$) for any two sets of inputs and randomnesses. In fact, in our construction, the simulator Sim^1 works by setting a random input to generate the first three rounds. Hence, for our construction, we require $T_{\mathsf{rMPC}_{(1-3)}}$-security for the following two distributions: $\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x}, z)$ and $\mathsf{Sim}^1(z)$. Further, for the two round non-malleable commitment, we have three parameters $T^{\mathsf{h}}_{\mathsf{nmcom}}$, $T^{\mathsf{nm}}_{\mathsf{nmcom}}$ and $\widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$. $T^{\mathsf{h}}_{\mathsf{nmcom}}$ and $T^{\mathsf{nm}}_{\mathsf{nmcom}}$ indicate that the hiding and non-malleability respectively of the non-malleable commitment hold against adversaries running in time $T^{\mathsf{h}}_{\mathsf{nmcom}}$ and $T^{\mathsf{nm}}_{\mathsf{nmcom}}$. $\widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$ refers to the running time of the rewinding extractor for the non-malleable commitment.

Theorem 10. *The described four round protocol is secure against malicious adversaries assuming the aforementioned leveled security of the primitives.*

All the primitives above with the desired security levels can be instantiated from sub-exponential DDH. The proof of theorem 10 can be found in appendix D.

6 Acknowledgements

The third author would like to thank Yuval Ishai for describing ideas for constructing a four-round semi-honest MPC protocol using randomizing polynomials.

The first author was supported by grant 360584 from the Simons Foundation. The second and the third authors were supported in part by a DARPA/ARL Safeware Grant W911NF-15-C-0213.

References

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 483–501, 2012.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[BHP17] Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. *IACR Cryptology ePrint Archive*, 2017:386, 2017.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 503–513, 1990.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 345–354, 2006.

[CGOS07] Nishanth Chandran, Vipul Goyal, Rafail Ostrovsky, and Amit Sahai. Covert multi-party computation. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 238–248. IEEE, 2007.

[COSV16] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Concurrent non-malleable commitments (and more) in 3 rounds. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 270–299, 2016.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. 4-round concurrent non-malleable commitments from one-way functions. In *CRYPTO*, 2017.

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 205–210, 1982.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th*

Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, pages 40–49. IEEE Computer Society, 2013.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 74–94, 2014.

[GGJS12] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Concurrently secure computation in constant rounds. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 99–116, 2012.

[GJ10] Vipul Goyal and Abhishek Jain. On the round complexity of covert computation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 191–200. ACM, 2010.

[GJO10] Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Password-authenticated session-key generation on the internet in the plain model. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 277–294, 2010.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 448–476, 2016.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, pages 291–304, 1985.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *STOC*, 1987.

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 695–704, 2011.

[Goy12] Vipul Goyal. Positive results for concurrently secure computation in the plain model. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 41–50, 2012.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1128–1141, 2016.

[GRRV14] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 41–50, 2014.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 294–304. IEEE, 2000.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. *IACR Cryptology ePrint Archive*, 2017:330, 2017.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 335–354, 2004.

[KOS03] Jonathan Katz, Rafail Ostrovsky, and Adam D. Smith. Round efficiency of multi-party computation with a dishonest majority. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 578–595, 2003.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. *IACR Cryptology ePrint Archive*, 2017:291, 2017.

[LPS17] Huijia Lin, Rafael Pass, and Pratik Soni. Two-round concurrent non-malleable commitment from time-lock puzzles. *IACR Cryptology ePrint Archive*, 2017:273, 2017.

[LS90] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 353–365, 1990.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 735–763, 2016.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 448–457, 2001.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 232–241, 2004.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 57–74, 2008.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 366–375, 2002.

[PW10] Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitments from sub-exponential one-way functions. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 638–655, 2010.

[Rab05] Michael O Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive*, 2005:187, 2005.

[Ros04] Alon Rosen. A note on constant-round zero-knowledge proofs for NP. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 191–202, 2004.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 543–553, 1999.

[vHL05] Luis von-Ahn, Nicholas Hopper, and John Langford. Covert two-party computation. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 513–522. ACM, 2005.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 531–540, 2010.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

A Definitions

We denote n to be the security parameter. Consider two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$. We denote $\mathcal{D}_0 \approx_c \mathcal{D}_1$ if $\mathcal{D}_0$ and $\mathcal{D}_1$ are computationally indistinguishable.

A.1 Oblivious Transfer

We recall the notion of oblivious transfer [Rab05, EGL82] below. We require that the oblivious transfer protocol satisfies *covert security* [vHL05, CGOS07, GJ10]. Intuitively, we require that the receiver's messages are computationally indistinguishable from a uniform distribution to a malicious sender. Similarly, we require that the sender's messages are computationally indistinguishable from a uniform distribution to a malicious receiver.

Definition 2 (Covert Oblivious Transfer)**.** *A 1-out-of-2 oblivious transfer (OT) protocol* OT *is a two party protocol between a sender and a receiver. A sender has two input bits* (b_0, b_1) *and the receiver has a choice bit* c*. At the end of the protocol, the receiver receives an output bit* b'*. We denote this process by* $b' \leftarrow \langle \mathsf{Sen}(b_0, b_1), \mathsf{Rec}(c)\rangle$*.*

We require that an OT protocol satisfies the following properties:

- **Correctness***: For every* $b_0, b_1, c \in \{0, 1\}$*, we have:*
$$\mathsf{Pr}[b_c \leftarrow \langle \mathsf{Sen}(b_0, b_1), \mathsf{Rec}(c)\rangle] = 1$$
- **Covert security against adversarial senders***: For all PPT senders* Sen^**, we require that the honest receiver's messages are computationally indistinguishable from uniform distribution.*
- **Covert security against adversarial receivers***: Suppose the input of the sender* (b_0, b_1) *is sampled from a distribution on* $\{0, 1\}^2$*. For all PPT receivers* Rec^**, we require that the honest sender's messages (computed as a function of* (b_0, b_1)*) are computationally indistinguishable.*

An oblivious transfer protocol satisfying the above definition was constructed in [vHL05] using [NP01].

Theorem 11 ([vHL05])**.** *Assuming decisional Diffie Helman assumption, there exists a two message 1-out-of-2 covert oblivious transfer protocol.*

A.2 Randomizing Polynomials

We first recall the definition of randomizing polynomials [IK00, AIK06]. Instead of considering the standard form of randomizing polynomials consisting of encode and decode algorithms, we instead consider a decomposable version where the circuit is first encoded as polynomials and decode algorithm gets as input evaluations of polynomials on input and randomness.

Definition 3 (Randomizing Polynomials)**.** *A randomizing polynomials scheme* $\mathsf{RP} = (\mathsf{CktE}, \mathsf{D})$ *for a family of circuits* $\mathcal{C}$ *has the following syntax:*

- *Encoding,* $\mathsf{CktE}(C)$*: On input circuit* $C \in \mathcal{C}$*, input* x*, it outputs polynomials* $p_1, \ldots, p_m$*.*

- *Decoding,* $\mathsf{D}(p_1(x;r), \ldots, p_m(x;r))$*: On input evaluations of polynomials* $p_1(x;r), \ldots, p_m(x;r)$*, it outputs the decoded value* α*.*

RP *is required to satisfy the following properties:*

- Correctness*: For every security parameter* $n \in \mathbb{N}$*, circuit* C *and input* x*,* $C(x) = \mathsf{D}(p_1(x;r), \ldots, p_m(x;r))$*, where (i)* $(p_1, \ldots, p_m) \leftarrow \mathsf{CktE}(C)$*, (ii)* r *is randomness sampled from uniform distribution.*

- Efficiency*: The typical efficiency we require is that the degree of the polynomials* $\{p_i\}$ *should be significantly smaller than the degree of the circuit* C*, where* $(p_1, \ldots, p_m) \leftarrow \mathsf{CktE}(C)$*.*

- Security*: For every PPT adversary* $\mathcal{A}$*, for large enough security parameter* $n \in \mathbb{N}$*, circuit* C *and input* x*, there exists a simulator* Sim *such that:*

$$\{(p_1(x;r), \ldots, p_m(x;r))\} \approx_c \left\{\mathsf{Sim}(1^n, 1^{|C|}, C(x))\right\},$$

where (i) $(p_1, \ldots, p_m) \leftarrow \mathsf{CktE}(C)$*, (ii)* r *is randomness sampled from uniform distribution.*

We define the **degree** *of randomizing polynomials to be* $\max_{C \in \mathcal{C}}\{\deg(p_i) \ : \ (p_1, \ldots, p_m) \leftarrow \mathsf{CktE}(C \in \mathcal{C})\}$*.*

We have the following theorem from [AIK06].

Theorem 12 ([AIK06])**.** *Assuming the existence of pseudorandom generators in* $\oplus\mathrm{L}/Poly$*, there exists a degree 3 randomizing polynomials for* $\mathcal{C}$*.*

A.3 Secure Multi-Party Computation

A secure multi-party computation protocol is a protocol executed by n number of parties $P_1, \cdots, P_\mathrm{n}$ for a n-party functionality F. We allow for parties to exchange messages simultaneously. In every round, every party is allowed to broadcast messages to all parties. A protocol is said to have k *rounds* if the number of rounds in the protocol is k. We require that at the end of the protocol, all the parties receive the output[8] $F(x_1, \ldots, x_\mathrm{n})$, where x_i is the i^{th} party's input. We formalize the security notion below.

Ideal World. We start by describing the ideal world experiment where n parties $P_1, \cdots, P_\mathrm{n}$ interact with an ideal functionality for computing a function F. An adversary may corrupt any subset $\mathcal{P}^\mathcal{A} \subset \mathcal{P}$ of the parties. We denote the honest parties by $\mathcal{H}$.

Inputs: Each party P_i obtains an initial input x_i. The adversary Sim is given auxiliary input z. Sim selects a subset of the parties $\mathcal{P}^\mathcal{A} \subset \mathcal{P}$ to corrupt, and is given the inputs x_k of each party $P_\mathrm{k} \in \mathcal{P}^\mathcal{A}$.

[8] We can also consider asymmetric functionalities where every party receives a different output. We don't discuss this in our work.

Sending inputs to trusted party: Each honest party P_{i} sends its input x_{i} to the trusted party. For each corrupted party $P_{\mathrm{i}} \in \mathcal{P}^{\mathcal{A}}$, the adversary may select any value x_{i}^* and send it to the ideal functionality.

Trusted party computes output: Let $x_1^*, ..., x_{\mathrm{n}}^*$ be the inputs that were sent to the trusted party. The trusted party sends $F(x_1^*, ..., x_{\mathrm{n}}^*)$ to the adversary who replies with either `continue` or `abort`. If the adversary's message is `abort`, then the trusted party sends $\perp$ to all honest parties. Otherwise, it sends the function evaluation $F(x_1^*, ..., x_{\mathrm{n}}^*)$ to all honest parties.

Outputs: Honest parties output all the messages they obtained from the ideal functionality. Malicious parties may output an arbitrary PPT function of the adversary's view.

The overall output of the ideal-world experiment consists of the outputs of all parties. For any ideal-world adversary Sim with auxiliary input $z \in \{0,1\}^*$, input vector $\vec{x}$, and security parameter n, we denote the output of the corresponding ideal-world experiment by

$$\mathsf{IDEAL}_{\mathsf{Sim},F}\Big(1^n, \vec{x}, z\Big).$$

Real World. The real world execution begins by an adversary $\mathcal{A}$ selecting any arbitrary subset of parties $\mathcal{P}^{\mathcal{A}} \subset \mathcal{P}$ to corrupt. The parties then engage in an execution of a real n-party protocol Π. Throughout the execution of Π, the adversary $\mathcal{A}$ sends all messages on behalf of the corrupted parties, and may follow an arbitrary polynomial-time strategy. In contrast, the honest parties follow the instructions of Π.

At the conclusion of all the update phases, each honest party P_{i} outputs all the outputs it obtained in the computations. Malicious parties may output an arbitrary PPT function of the view of $\mathcal{A}$.

For any adversary $\mathcal{A}$ with auxiliary input $z \in \{0,1\}^*$, input vector $\vec{x}$, and security parameter n, we denote the output of the MPC protocol Π by

$$\mathsf{REAL}_{\mathcal{A},\Pi}\Big(1^n, \vec{x}, z\Big).$$

Security Definition. We say that a protocol Π is a secure protocol if any adversary, who corrupts a subset of parties and runs the protocol with honest parties, gains *no information* about the inputs of the honest parties beyond the protocol output.

Definition 4. *A protocol* Π *is a secure* n*-party protocol computing* F *if for every PPT adversary* $\mathcal{A}$ *in the real world, there exists a PPT adversary* Sim *corrupting the same parties in the ideal world such that for every initial input vector* $\vec{x}$*, every auxiliary input* z*, it holds that*

$$\mathsf{IDEAL}_{\mathsf{Sim},F}\Big(1^n, \vec{x}, z\Big) \approx_c \mathsf{REAL}_{\mathcal{A},\Pi}\Big(1^n, \vec{x}, z\Big).$$

A.4 Non-malleable Commitments

Let $\Pi = \langle C, R \rangle$ be a statistically binding commitment scheme. Consider man-in-the-middle (MiM) adversaries that are participating in one left and one right sessions in which k commitments take place. We compare between a MiM and a simulated execution. In the MiM execution, the adversary $\mathcal{A}$, with auxiliary information z, is participating in one left and one right sessions. In the left sessions the MiM adversary interacts with C receiving commitments to value m using identities $\mathtt{id}$ of its choice. In the right session $\mathcal{A}$ interacts with R, attempting to commit to a related value $\tilde{m}$ again using identities $\tilde{\mathtt{id}}$ of its choice. If any the right commitment is invalid, or undefined, its value is set to $\perp$. If $\tilde{\mathtt{id}} = \mathtt{id}$, set $\tilde{m} = \perp$ (i.e., any commitment where the adversary uses the same identity as that of honest senders is considered invalid). Let

$$\mathsf{mim}_{\Pi}^{\mathcal{A},m}(z)$$

denote the random variable that describes the values $\tilde{m}$ and the view of $\mathcal{A}$, in the above experiment.

In the simulated execution, an efficient simulator Sim directly interacts with R. Let

$$\mathsf{sim}_{\Pi}^{\mathsf{Sim}}(1^n, z)$$

denote the random variable describing the value $\tilde{m}$ committed by Sim, and the output view of Sim; whenever the view contains the same identity as that identity of the left session, $\tilde{m}$ is set to $\perp$.

Definition 5 (non-malleable commitment scheme)**.** *A commitment scheme is non-malleable with respect to commitment if, for every* PPT *parallel MiM adversary* $\mathcal{A}$*, there exists a* PPT *simulator* Sim *such that for all* m *the following ensembles are computationally indistinguishable:*

$$\{\mathsf{mim}_{\Pi}^{\mathcal{A},m}(z)\}_{n\in\mathbb{N},z\in\{0,1\}^*} \approx \{\mathsf{sim}_{\Pi}^{\mathsf{Sim}}(1^n, z)\}_{n\in\mathbb{N},z\in\{0,1\}^*}$$

Non-malleable Commitment Schemes. In our MPC constructions, we make use of two different non-malleable commitment schemes:

- *Four-round public-coin extractable NMCOM.* The first commitment scheme we use is a four-round NMCOM scheme that is public-coin w.r.t. the receiver, and also supports extraction of the committed value in polynomial time. Such a commitment scheme was constructed from CRHFs in [GRRV14].

- *Two-round (private-coin) extractable NMCOM.* We also use a two-round (private-coin) NMCOM scheme that supports rewinding-based extraction of the committed value, possibly in sub-exponential time. We further require that the commitment scheme achieves hiding (and non-malleability) even against adversaries that run in time $T' >> T$, where T is the running time of the extractor. Such a commitment scheme was constructed from DDH by [KS17].[9]

  This commitment scheme is used in the construction of our four-round MPC protocol in Section 5. Our construction and proof can be modified to instead use a three-round input delayed public-coin non-malleable commitment scheme that supports extraction of the committed value in polynomial time. (We in fact only require successful extraction with polynomial probability.) Such commitment schemes were constructed in [GPR16, COSV16].

A.5 Input Delayed Non-malleable Zero Knowledge

Let $\Pi_{\mathsf{nmzk}} = \langle P, V\rangle$ be a input delayed interactive argument system for an NP-language L with witness relation Rel_L. Consider a PPT MiM adversary $\mathcal{A}$ that is simultaneously participating in one left session and one right session. Before the execution starts, both P, V and $\mathcal{A}$ receive as a common input the security parameter n, and $\mathcal{A}$ receives as auxiliary input $z \in \{0,1\}^*$.

In the left session $\mathcal{A}$ interacts with P using identity $\mathtt{id}$ of his choice. In the right session, $\mathcal{A}$ interacts with V, using identity $\widetilde{\mathtt{id}}$ of his choice. In the left session, before the last round of the protocol, P gets the statement x. Also, in the right session $\mathcal{A}$, during the last round of the protocol selects the statement $\tilde{x}$ to be proved and sends it to V. Let $\mathsf{View}^{\mathcal{A}}(1^n, z)$ denote a random variable that describes the view of $\mathcal{A}$ in the above experiment.

Definition 6 (Input Delayed NMZK)**.** *A input delayed argument system* $\Pi_{\mathsf{nmzk}} = \langle P, V\rangle$ *for* NP*-language* L *with witness relation* $\mathsf{Rel_L}$ *is Non-Malleable Zero Knowledge (NMZK) if for any MiM adversary* $\mathcal{A}$ *that participates in one left session and one right session, there exists a* PPT *machine* $\mathsf{Sim}(1^n, z)$ *such that*

[9] A different two-round non-malleable commitment scheme (without the latter property) was recently constructed by [LPS17] based on time-lock puzzles.

1. *The probability ensembles* $\{\mathsf{Sim}^1(1^n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$ *and* $\{\mathsf{View}^{\mathcal{A}}(1^n, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ *are computationally indistinguishable over* n*, where* $\mathsf{Sim}^1(1^n, z)$ *denotes the first output of* $\mathsf{Sim}(1^n, z)$*.*

2. *Let* $z \in \{0,1\}^*$ *and let* $(\mathsf{View}, \tilde{w})$ *denote the output of* $\mathsf{Sim}(1^n, z)$*. Let* $\tilde{x}$ *be the right-session statement appearing in* View *and let* $\mathtt{id}$ *and* $\tilde{\mathtt{id}}$ *be the identities of the left and right sessions appearing in* View*. If the right session is accepting and* $\mathtt{id} \neq \tilde{\mathtt{id}}$*, then* $\mathsf{Rel}_{\mathsf{L}}(\tilde{x}, \tilde{w}) = 1$*.*

A.5.1 COSV NMZK

Here, we describe the four round input delayed NMZK protocol Π_{COSV} [COSV17]. We start by recalling the main cryptographic primitives used in their protocol:

1. a 4-round public-coin extractable one-one NM commitment scheme $\Pi_{\mathsf{nmex}} = \langle C_{\mathsf{nmex}}, R_{\mathsf{nmex}}\rangle$; [COSV17] instantiates this using the non-malleable commitment scheme constructed in the same paper.

2. a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$;

3. an input delayed adaptive-input statistical WIAoK protocol $\mathsf{sLS} = (P_{\mathsf{sLS}}, V_{\mathsf{sLS}})$ for the language

$$\begin{aligned}\Lambda = \Big\{ &(\tau_{\mathrm{k}} = (\pi^1_{\mathsf{nmex}}, \pi^2_{\mathsf{nmex}}, \pi^3_{\mathsf{nmex}}, \pi^4_{\mathsf{nmex}}), \mathtt{id}, \mathsf{vk}, x, s_1) : \exists (s_0, \mathsf{dec}, \mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2) \\ &\text{s.t. } \Big((R_{\mathsf{nmex}} \text{ on input } (\tau, w, \mathsf{dec}, \mathtt{id}) \text{ accepts } s_0 \text{ as decommitment of } \tau \\ &\quad \mathtt{AND}\ (x, s_0 \oplus s_1) \in \mathsf{Rel}_{\mathsf{L}})\ \mathtt{OR}\ (\mathsf{Ver}(\mathsf{vk}, \mathsf{msg}_1, \sigma_1) = 1\ \mathtt{AND}\ \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}_2, \sigma_2) = 1 \\ &\quad \mathtt{AND}\ \mathsf{msg}_1 \neq \mathsf{msg}_2) \Big) \Big\}\end{aligned}$$

   that is adaptive-input statistical WI and adaptive-input AoK for the corresponding relation Rel_{Λ}.

The Protocol. We now describe the COSV NMZK protocol.

Common input: security parameter n, the instance length ℓ of sLS and P_{nmzk}'s identity $\mathtt{id} \in \{0,1\}^n$, and the instance x is available only at the last round.

Private input of P_{nmzk}*:* w s.t. $(x, w) \in \mathsf{Rel}_L$ available only in the last round.

1. $V_{\mathsf{nmzk}} \to P_{\mathsf{nmzk}}$

   (a) $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^n)$.

   (b) $\pi^1_{\mathsf{sLS}} \leftarrow V_{\mathsf{sLS}}(1^n, \ell)$.

   (c) $\pi^1_{\mathsf{nmex}} \leftarrow R_{\mathsf{nmex}}(1^n, \mathtt{id})$.

   (d) Send $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{nmex}})$ to P_{nmzk}

2. $P_{\mathsf{nmzk}} \to V_{\mathsf{nmzk}}$

   (a) $\pi^2_{\mathsf{sLS}} \leftarrow P_{\mathsf{sLS}}(1^n, \ell, \pi^1_{\mathsf{sLS}})$.

   (b) $s_0 \xleftarrow{\$} \{0,1\}^{|w|}$.

   (c) $\pi^2_{\mathsf{nmex}} \leftarrow C_{\mathsf{nmex}}(1^n, \mathtt{id}, \pi^2_{\mathsf{nmex}}, s_0)$.

   (d) $\mathsf{msg} \xleftarrow{\$} \{0,1\}^n$.

   (e) Send $(\pi^2_{\mathsf{sLS}}, \pi^2_{\mathsf{nmex}}, \mathsf{msg})$ to V_{nmzk}.

3. $V_{\mathsf{nmzk}} \to P_{\mathsf{nmzk}}$

   (a) $\pi^3_{\mathsf{sLS}} \leftarrow V_{\mathsf{sLS}}(\pi^2_{\mathsf{sLS}})$.
   (b) $\pi^3_{\mathsf{nmex}} \leftarrow R_{\mathsf{nmex}}(\pi^2_{\mathsf{nmex}})$.
   (c) $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$.
   (d) Send $(\pi^3_{\mathsf{sLS}}, \pi^3_{\mathsf{nmex}}, \sigma)$ to P_{nmzk}.

4. $P_{\mathsf{nmzk}} \rightarrow V_{\mathsf{nmzk}}$
   (a) If $\mathsf{Ver}(\mathsf{vk}, \mathsf{msg}, \sigma) \neq 1$ then abort, else continue.
   (b) Set $s_1 := w \oplus s_0$.
   (c) $(\mathsf{dec}, \pi^4_{\mathsf{nmex}}) \leftarrow C_{\mathsf{nmex}}(\pi^2_{\mathsf{nmex}})$.
   (d) Set $x_{\mathsf{sLS}} := (\pi^1_{\mathsf{nmex}}, \pi^2_{\mathsf{nmex}}, \pi^3_{\mathsf{nmex}}, \pi^4_{\mathsf{nmex}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} := (s_0, \mathsf{dec}, \bot, \bot, \bot, \bot)$.
   (e) $\pi^4_{\mathsf{sLS}} \leftarrow P_{\mathsf{sLS}}(\pi^3_{\mathsf{sLS}}, x_{\mathsf{sLS}}, w_{\mathsf{sLS}})$.
   (f) Send $(\pi^4_{\mathsf{sLS}}, \pi^4_{\mathsf{nmex}}, s_1)$ to V_{nmzk}.

5. V_{nmzk}: Set $x_{\mathsf{sLS}} := (\pi^1_{\mathsf{nmex}}, \pi^2_{\mathsf{nmex}}, \pi^3_{\mathsf{nmex}}, \pi^4_{\mathsf{nmex}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and accept iff $V_{\mathsf{sLS}}(x_{\mathsf{sLS}}, \pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}, \pi^4_{\mathsf{sLS}}) = 1$.

Modified COSV NMZK. We use a slight modification of the above protocol in our MPC constructions. Specifically, we change the value committed in the non-malleable commitment, as well as the language Λ:

- The mask s_1 is set to the same value as s_0 such that $s_0 \oplus s_1 = 0$. (That is, the honest prover commits to 0.)

- The language for the WIAoK is changed to

$$\Lambda = \Big\{ \widetilde{x} = \big(x, (\tau_{\mathsf{k}} = (\pi^1_{\mathsf{nmex}}, \pi^2_{\mathsf{nmex}}, \pi^3_{\mathsf{nmex}}, \pi^4_{\mathsf{nmex}}), \mathtt{id}, \mathsf{vk}, s_1)\big) : \exists (w, s_0, \mathsf{dec}, \mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2) \\ \text{s.t. } \Big(((x, w) \in \mathsf{Rel}_{\mathsf{L}}) \;\mathtt{OR}\; (R_{\mathsf{nmex}} \text{ on input } (\tau, s_0, \mathsf{dec}, \mathtt{id}) \text{ accepts } s_0 \text{ as decommitment of} \\ \tau \;\mathtt{AND}\; s_0 \oplus s_1 = (\mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2) \;\mathtt{AND}\; \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}_1, \sigma_1) = 1 \;\mathtt{AND}\; \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}_2, \sigma_2) = 1 \\ \mathtt{AND}\; \mathsf{msg}_1 \neq \mathsf{msg}_2) \Big) \Big\}$$

  i.e. either the statement x is in the language L, or a "trapdoor witness" $(\mathsf{msg}_1, \mathsf{msg}_2, \sigma_1, \sigma_2)$ was committed to inside the non-malleable commitment.

  Notation: For any instance $\widetilde{x}$ in Λ, we refer to the first part of the statement, i.e., x as the *honest statement*. We refer to the second part of $\widetilde{x}$, i.e., $(\tau_{\mathsf{k}} = (\pi^1_{\mathsf{nmex}}, \pi^2_{\mathsf{nmex}}, \pi^3_{\mathsf{nmex}}, \pi^4_{\mathsf{nmex}}), \mathtt{id}, \mathsf{vk}, s_1)$, as the *trapdoor statement*.

Theorem 13 ([COSV17]). *Assuming CRHFs, the above protocol is a secure NMZK.*

In terms of concrete instantiations, CRHFs can be instantiated from DL, and hence from DDH.

A.6 Extractable Commitment Scheme

We will use a variant of a simple challenge-response based extractable statistically-binding string commitment scheme $\langle C, R \rangle$ that has been used in several prior works, most notably [PRS02, Ros04]. We note that in contrast to [PRS02] where a multi-slot protocol was used, here (similar to [Ros04]), we only need a one-slot protocol.

Protocol $\langle C, R \rangle$. Let $\mathsf{com}(\cdot)$ denote the commitment function of a non-interactive perfectly binding string commitment scheme which requires the assumption of injective one-way functions for its construction. Let n denote the security parameter. The commitment scheme $\langle C, R \rangle$ is described as follows.

Commit Phase:

1. To commit to a string $\mathtt{str}$, C chooses $k = \omega(\log(n))$ independent random pairs $\{\alpha_i^0, \alpha_i^1\}_{i=1}^k$ of strings such that $\forall i \in [k]$, $\alpha_i^0 \oplus \alpha_i^1 = \mathtt{str}$; and commits to all of them to R using com. Let $B \leftarrow \mathsf{com}(\mathtt{str})$, and $A_i^0 \leftarrow \mathsf{com}(\alpha_i^0)$, $A_i^1 \leftarrow \mathsf{com}(\alpha_i^1)$ for every $i \in [k]$.
2. R sends k uniformly random bits $v_1, \ldots, v_n$.
3. For every $i \in [k]$, if $v_i = 0$, C opens A_i^0, otherwise it opens A_i^1 to R by sending the appropriate decommitment information.

Open Phase: C opens all the commitments by sending the decommitment information for each one of them.

For our construction, we require a modified extractor for the extractable commitment scheme. The standard extractor returns the value $\mathtt{str}$ that was committed to in the scheme. Instead, we require that the extractor return i, and the openings of A_i^0 and A_i^1. This extractor can be constructed easily, akin to the standard extractor for the extractable commitment scheme.

This completes the description of $\langle C, R \rangle$.

"Rewinding secure" Commitment Scheme. Due to technical reasons, we will use a minor variant, denoted $\langle C', R' \rangle$, of the above commitment scheme which will is "rewinding secure." Protocol $\langle C', R' \rangle$ is the same as $\langle C, R \rangle$, except that for a given receiver challenge string, the committer does not "open" the commitments, but instead simply reveals the appropriate committed values (without revealing the randomness used to create the corresponding commitments). More specifically, in protocol $\langle C', R' \rangle$, on receiving a challenge string $v_1, \ldots, v_n$ from the receiver, the committer uses the following strategy: for every $i \in [k]$, if $v_i = 0$, C' sends α_i^0, otherwise it sends α_i^1 to R'. Note that C' does not reveal the decommitment values associated with the revealed shares.

The scheme is rewinding secure because we can respond to queries from the adversary (for the commitment scheme) when we need to rewind it, and the commitment scheme is exposed to an external challenger. This follows from the fact that we can send random messages in the third round when the adversary makes a different second round query.

When we use $\langle C', R' \rangle$ in our main construction, we will require the committer C' to prove the "correctness" of the values (i.e., the secret shares) it reveals in the last step of the commitment protocol. In fact, due to technical reasons, we will also require the the committer to prove that the commitments that it sent in the first step are "well-formed". Below we formalize both these properties in the form of a *validity* condition for the commit phase.

Proving Validity of the Commit Phase. We say that commit phase between C' and R' is *well formed* with respect to a value $\hat{\mathtt{str}}$ if there exist values $\{\hat{\alpha}_i^0, \hat{\alpha}_i^1\}_{i=1}^k$ such that:

1. For all $i \in [k]$, $\hat{\alpha}_i^0 \oplus \hat{\alpha}_i^1 = \hat{\mathtt{str}}$, and
2. Commitments B, $\{A_i^0, A_i^1\}_{i=1}^k$ can be decommitted to $\hat{\mathtt{str}}$, $\{\hat{\alpha}_i^0, \hat{\alpha}_i^1\}_{i=1}^k$ respectively.
3. Let $\bar{\alpha}_1^{v_1}, \ldots, \bar{\alpha}_k^{v_k}$ denote the secret shares revealed by C in the commit phase. Then, for all $i \in [k]$, $\bar{\alpha}_i^{v_i} = \hat{\alpha}_i^{v_i}$.

The lemma below states that $\exists$ an extractor E that extracts the correct committed value with overwhelming probability if the commitment is well formed. This lemma is implicit in [Ros04, PRS02].

Lemma 1. *If the validity condition for the commitment protocol holds, then E fails to extract the committed value with only negligible probability.*

B Proofs From Section 3

B.1 Proof of Theorem 4

We only need to argue about the output of P_3. From the correctness of OT_{12}, it follows that P_1 recovers $x_1x_2 + r'_2$ in Round 2. From the correctness of OT_{23}, it follows that P_3 recovers $\alpha'' = x_3r'_2 + r_2$. Finally, from the correctness of OT_{13}, it follows that P_3 recovers $\alpha' = (x_1x_2 + r'_2)x_3 + r_1$. Note that $\alpha' + \alpha'' = x_1x_2x_3 + r_1 + r_2$, as desired.

B.2 Proof of Theorem 5

We consider all maximal sets of corruptions and argue security. In each case, we construct a simulator that sends pseudorandom messages in the first two rounds.

P_1 and P_2 are corrupted: In this case, simulator (Sim^1)essentially runs honest P_3 algorithm but with input $x_3 = 0$. In the final (fourth) round, the simulator (Sim^2) upon receiving as input $((\mathsf{3MULT}((x_1, r_1); (x_2, r_2); x_3) = (\alpha_1, \alpha_2, \alpha_3)), (x_1, r_1), (x_2, r_2))$, it outputs α_3.

The the security requirement of the robust semi honest MPC follows from that of oblivious transfer protocol and the covert security property of OT. The covertness gives us the desired joint distribution

P_1 and P_3 are corrupted: The simulator runs the honest P_2 with input $x_2 = 0$. Note that the output of P_1 and P_3 are r_1 and $r_1 + r_2$ respectively. In the final round, the simulator upon receiving as input $(\mathsf{3MULT}(\ (x_1, r_1); (x_2, r_2); x_3) = (\alpha_1, \alpha_2, \alpha_3)),\ (x_1, r_1), x_3)$, outputs $\alpha_3 + r_1$ (which is of the form $x_1x_2x_3 + r_2$).

The the security requirement of the robust semi honest MPC follows from that of oblivious transfer protocol and the covert security property of OT. The covertness gives us the desired joint distribution

P_2 and P_3 are corrupted: This is symmetrical to the previous case. Following a similar argument we result in the simulator outputting $x_1x_2x_3 + r_1$, P_2 outputs r_2 and P_3 outputs $r_1 + r_2$.

B.3 Proof of Theorem 6

Let the additive shares of 0 distributed by P_i be $\{s_0^{i,j}\}_{j\in[n]}$. Consider a term t in the expansion of p. Without loss of generality, let $\mathbf{y}_i, \mathbf{y}_j$ and $\mathbf{y}_k$ be the variables in the expansion of t. From the correctness of $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$, it follows that at the end of third round, P_i, P_j and P_k have shares of $x_ix_jx_k$. Denote these additive shares by α_i^t, α_j^t and α_k^t. At the end of the protocol, the share computed by P_i is total sum of $\sum_{i,j} s_0^{i,j}$ and the sum of shares corresponding to every term t in p. Observe that $\sum_{i,j} s_0^{i,j}$ is 0 and the sum of shares corresponding to every term t in p is $p(x_1, \ldots, x_n)$.

B.4 Proof of Theorem 7

Suppose S is the set of corrupted parties controlled by adversary $\mathcal{A}$. We describe a simulator Sim that simulates the corrupted parties in S.

Description of Simulator. Recall that for every polynomial p, an instantiation of $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ is executed. Consider a term t in the expansion of p. We look at two cases:

- Case 1: t contains one variable associated with party not in S and contains another variable with party associated with a party in S: In this case, Sim executes the simulator, denoted by $\mathsf{Sim}_{\mathsf{3MULT}}$, of $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$. Recall that in the fourth round, $\mathsf{Sim}_{\mathsf{3MULT}}$ takes as input the output of the functionality as well as the inputs and randomness of all the adversaries. In particular, Sim internally computes the functionality by setting the inputs of all the honest parties to 0 and this is input to $\mathsf{Sim}_{\mathsf{3MULT}}$.

- Case 2: t contains only variables associated with parties in S: The protocol associated with t is executed solely by adversarial parties.

The only remaining case was when t does not contain any variable associated with a party in S. In this case, the simulator Sim upon receiving the input $(p(x_1,\ldots,x_n),\{x_i,r_i\}_{i\in S})$, generates the final round messages as follows: it computes $\alpha = p(x_1,\ldots,x_n)-\beta$, where β is the summation of all the terms in the expansion of p such that these terms contain only variables associated with parties in S. The simulator then computes the final round messages of all the honest parties to be shares of the value α.

The above described simulator satisfies Definition 1 from the fact that $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$ is a robust semi-honest MPC, and the fact that the last messages generated by the simulator is distributed identically to the last messages generated by the honest parties in the real world.

B.5 Proof of Theorem 8

We describe the simulator below.

Description of Simulator Sim. Let C be the circuit implementing the functionality F. Execute $\mathsf{CktE}(C)$ to get $(p_1,\ldots,p_m)$. Execute the simulator $\mathsf{Sim}_{\mathsf{3POLY}\{p_i\}}$ for every $i\in[m]$ for the first three rounds.

In the final round, Sim receives as input $(F(x_1,\ldots,x_n),\{x_i,r_i\}_{i\in S})$. It first executes the simulator of RP on input $F(x_1,\ldots,x_n)$ to obtain $(\beta_1,\ldots,\beta)$. It then executes the final round of $\mathsf{Sim}_{\mathsf{3POLY}\{p_i\}}$ on input $((\beta_1,\ldots,\beta_n),\{x_i,r_i\}_{i\in S})$ for every $i\in[m]$. Denote the outputs of the simulators to be $\overrightarrow{\alpha}=(\alpha_1,\ldots,\alpha_m)$. Output $\overrightarrow{\alpha}$.

We now prove that the simulator satisfies definition 1 by hybrid argument.

Hybrid Hyb_0: This corresponds to the real world.

Hybrid Hyb_1: In this hybrid, execute the simulator $\mathsf{Sim}_{\mathsf{3POLY}\{p_i\}}$ for every $i\in[m]$ for the first three rounds. In the final round, execute $\mathsf{Sim}_{\mathsf{3POLY}\{p_i\}}$ on input $(\beta_i,\{x_i,r_i\}_{i\in S})$, where β_i is computed by evaluating p_i honestly on the inputs of all the parties. The output of Sim is just a concatenation of outputs of $\mathsf{Sim}_{\mathsf{3POLY}_{p_i}}$ for every i.

The indistinguishability of Hyb_0 and Hyb_1 follows from the security of $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$.

Hybrid Hyb_2: This hybrid corresponds to the ideal world.

Observe that in Hyb_1, the $\{\beta_i\}$ input to $\mathsf{Sim}_{\mathsf{3POLY}\{p_i\}}$ is identically distributed to the encoding of the circuit according to RP. We can now invoke the security of RP to argue the indistinguishability of Hyb_2 and Hyb_3

Thus, from the indistinguishability of the hybrids and the fact that $\mathsf{Sim}_{\mathsf{3POLY}\{p\}}$ satisfies Definition 1, Π_{sh}^{F} is a robust semi honest MPC.

B.6 Special Rewinding property

We highlight a special property of our constructed four round robust semi-honest MPC, which we shall refer to as the "special rewinding" property. This will be useful for the proof of our five round construction. Roughly, the property states that the last round of the robust semi honest MPC can be simulated without knowledge of the input and randomness used in the first two rounds. For our construction, a random string will be indistinguishable from an honestly generated message in the third round, when the view is restricted to the first three rounds.

Claim 1. *Let h_i (resp. a_i) denote all the messages sent by the honest (resp. adversarial) parties in the i-th round. Then the following joint distributions are indistinguishable: $(h_1, a_1, h_2, a_2, h_3)$ and $(h_1, a_1, h_2, a_2', h_3')$ where h_3' generated without knowledge of random coins and inputs used to compute h_1 and h_2.*

Proof sketch. Our construction of the robust semi honest MPC relies on the computation of m randomized polynomials. We argue that the property holds for any monomial, and this can be extended to the case of the polynomials. While there are common inputs across various monomials and polynomials, each monomial samples independent randomness for its computation and this suffices to let us argue them separately. The main property of our underlying construction we will use is of the security of the OT.

From the construction of $\Pi_{\mathsf{sh}}^{\mathsf{3POLY}\{p\}}$, which internally invokes $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$, each player has a specific role for a given monomial: (i) it is not involved; (ii) involved and has a predefined role of either P_1, P_2 or P_3. Where P_1, P_2 and P_2 have roles as described in $\Pi_{\mathsf{sh}}^{\mathsf{3MULT}}$. The first case is trivial since we don't need to send anything. Let us consider the 3 other cases. If the player has the role of P_2 or P_3, then by construction it is not required to send anything in the last round. If the player has the role of P_1, then it has to respond to an OT message in round 3. The message sent is a function of the prior messages, but is masked by r_1 which has not been used prior to round 3 and is chosen independently for the given monomial. Thus from the OT sender security, we can pick u and r_1 randomly to construct the third round message. The security holds even if P_2 and P_3 collude and know the value of u since the mask is chosen randomly and used for the first time.

In our proof, we shall need this property to respond to (potentially different) queries sent by the adversary in the second round while rewinding when we argue security via the robust semi-honest MPC.

C Proof of Theorem 9

We present the proof for our five round construction below. Before we proceed to the simulator, we discuss a few properties of the underlying primitives that we will need:

- Recall that simulator for the robust semi honest MPC consists of two parts. The first part, $\mathsf{Sim}_{\mathsf{rMPC}}^1$, simulates the first three rounds of the robust semi honest MPC without requiring inputs or outputs of the adversary. The second part, $\mathsf{Sim}_{\mathsf{rMPC}}^2$, when given the inputs, random tape and outputs a simulated transcript of the last round that is consistent with the input and randomness. Additionally, note that this simulation succeeds as long as the adversary behaved honestly in the first three rounds of the robust semi honest MPC.

- The extractor for the 3 round "rewinding secure" extractable commitment works by rewinding the second and third round polynomial number of times. From Lemma 1, we know that if the commitments are well formed, extraction fails with only negligible probability.

- The simulator of the NMZKs works by extracting a trapdoor. Specifically, it rewinds the second and third round polynomial number of times to get signatures for two distinct messages. Further, this extraction fails only with negligible probability if the adversary does not abort with non-negligible probability.

- Combining the above two properties, we see that the rewindings of NMZK and the extractable commitment are "composable" because they rewind in the same rounds in our MPC protocol.

- To extract the value in the four round non-malleable commitment within the NMZK, we rewind in the third and fourth round. This will be useful will arguing the proofs in the hybrids.

C.1 Description of the Simulator

We describe the ideal world simulator Sim below. We shall denote the set of honest players by $\mathcal{H}$ and the set of corrupted players by $\mathcal{P}^{\mathcal{A}}$.

1. The first three rounds of protocol are simulated as follows:
   - For the robust semi honest MPC, since $\mathsf{Sim}^{1}_{\mathsf{rMPC}}$ doesn't require any input or output to simulate the first three rounds, we use it directly to obtain $\left\{m^{1}_{\mathrm{i}}, m^{2}_{\mathrm{i}}, m^{3}_{\mathrm{i}}\right\}_{P_{\mathrm{i}}\in\mathcal{H}}$.
   - For simulating proofs for the NMZKs, we deal with three different cases:
     (a) For proofs from the adversary, the honest player acts as a verifier. In this case, fix a random tape for the verifier and respond honestly to adversary queries.
     (b) For proofs within honest players, we fix the random tape for the verifiers and thus can trivially compute the trapdoor in the NMZKs for both languages using the verifier's random tape.
     (c) For proofs from honest players to the adversary, we run the simulator $\mathsf{Sim}_{\mathsf{nmzk}}$ a to simulate the first three rounds. This internally rewinds polynomial many times to obtain the trapdoors. If the extractor fails, output $\bot_{\mathsf{nmzk}}$ and abort.

     This gives us $\left\{\pi^{\mathsf{j}}_{\mathsf{nmzk}_{\mathsf{i}}}\right\}_{\mathrm{j}\in\{1,2,3\},P_{\mathrm{i}}\in\mathcal{H}}$ and the extracted trapdoors.
   - For the extractable commitment, we deal with two cases:
     (a) For commitments from the honest players to the adversary, we just commit to the '0' string. We do this for commitments within the honest players as well.
     (b) For commitments where the honest players are recipients, run the extractor to send responses and extract the values inside the commitments. If extractor fails, output $\bot_{\mathsf{ecom}}$ and abort.

     This gives us $\left\{\pi^{\mathsf{j}}_{\mathsf{ecom}_{\mathsf{i}}}\right\}_{\mathrm{j}\in\{1,2,3\},P_{\mathrm{i}}\in\mathcal{H}}$ and the extracted commitments.
   - Only the first round of the WIPoK overlaps with the first three rounds of the protocol. We behave honestly for both the first and second rounds of the WIPoK by fixing a random tape for the honest players. Even though the second round overlaps with the fourth round of the protocol, we group it here for simplicity.

   As noted earlier, the rewinding performed within the NMZK simulator and the extractor for extractable commitments work in the same rounds and can be done for each without affecting the other.

2. Simulate the last round of the NMZK for L in two steps.
   - For proofs from the honest parties to the adversary, use $\mathsf{Sim}_{\mathsf{nmzk}}$ with inputs $\left\{\pi^{\mathsf{j}}_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}}\right\}_{\mathrm{j}\in\{1,2,3\},\ P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}},P_{\mathrm{i}}\in\mathcal{H}}$ and the trapdoors obtained earlier to compute
   $$\left\{\pi^{4}_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}}\right\}_{P_{\mathrm{k}}\ \in\mathcal{P}^{\mathcal{A}},\ P_{\mathrm{i}}\ \in\mathcal{H}}.$$
   - For proofs within honest parties, the trapdoor is trivially known to the adversary and thus use $\left\{\pi^{\mathsf{j}}_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}}\right\}_{\mathrm{j}\in\{1,2,3\},\ P_{\mathrm{k}},P_{\mathrm{i}}\in\mathcal{H}}$ to construct
   $$\left\{\pi^{4}_{\mathsf{nmzk}_{\mathsf{i}\to\mathsf{k}}}\right\}_{P_{\mathrm{k}},\ P_{\mathrm{i}}\ \in\mathcal{H}}.$$

This gives us the required $\left\{\pi^4_{\mathsf{nmzk}_\mathrm{i}}\right\}_{P_\mathrm{i}\ \in\mathcal{H}}$.

On receiving the proofs from the adversary check if all the received proofs are valid i.e. verify if $\{\pi^\mathrm{j}_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}\}_{\mathrm{j}\in\{1,2,3,4\},\ P_\mathrm{k}\ \in\mathcal{P}^\mathcal{A},\ P_\mathrm{i}\ \in\mathcal{H}}$ are valid proofs in L (This is equivalent to checking if all proofs in the protocol verify). If the check fails, send `abort` to the ideal functionality.

3. We perform an additional check before we obtain the final round of the robust semi honest MPC. Given $\vec{m}^1, \vec{m}^2, \vec{m}^3, \{(x_k, r_k)\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}$, we check if the adversary has followed the computation in the first three rounds correctly. If the check fails we output $\bot^1_{\mathsf{rMPC}}$ and abort. It is implicit that the proofs for L have verified prior to this step.

4. Send the extracted inputs $\{x_k\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}$ to the ideal functionality to obtain the output y.

   Compute the final round (of all players) of the robust semi honest MPC as

$$\left\{m^4_\mathrm{i}\right\}_{P_\mathrm{i}\in\mathcal{P}} \leftarrow \mathsf{Sim}^2_{\mathsf{rMPC}}\left(\vec{m}^1, \vec{m}^2, \vec{m}^3, \{x_\mathrm{k}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}, \{r_\mathrm{k}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}, y\right).$$

   Additionally, simulate the last round of the WIPoK for L_{WIPoK}. This is done in two steps

   - For proofs from the honest parties to the adversary we use the trapdoors obtained earlier to compute the proof with the trapdoor witness

$$\left\{\pi^4_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}\right\}_{P_\mathrm{k}\ \in\mathcal{P}^\mathcal{A},\ P_\mathrm{i}\ \in\mathcal{H}}.$$

   - For proofs within honest parties, the trapdoor is trivially known and thus use $\left\{\pi^\mathrm{j}_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{j}\in\{1,2,3\},\ P_\mathrm{k},P_\mathrm{i}\in\mathcal{H}}$ to construct

$$\left\{\pi^4_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}\right\}_{P_\mathrm{k},\ P_\mathrm{i}\ \in\mathcal{H}}.$$

   This gives us the required $\left\{\pi^4_{\mathsf{WIPoK}_\mathrm{i}}\right\}_{P_\mathrm{i}\ \in\mathcal{H}}$.

5. On receiving the proofs from the adversary check if all the received proofs are valid, i.e. verify if $\{\pi^\mathrm{j}_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}}\}_{\mathrm{j}\in\{1,2,3,4\},\ P_\mathrm{k}\ \in\mathcal{P}^\mathcal{A},\ P_\mathrm{i}\ \in\mathcal{H}}$ are valid proofs in L_{WIPoK}. If the check fails, send `abort` to the ideal functionality.

   Otherwise, on receiving $\left\{m^{*4}_\mathrm{k}\right\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}$ from the adversary, we check if it matches the transcript simulated by $\mathsf{Sim}^2_{\mathsf{rMPC}}$ earlier. If not, but the proofs above have verified output $\bot^2_{\mathsf{rMPC}}$ and abort. Else send `continue` to the ideal functionality.

C.2 Description of the hybrids

We prove security via a sequence of hybrids H_0 to H_6 described below, where H_0 is the real execution and H_6 is the ideal execution.

Random variables. We introduce the following random variables and their indistinguishability will be argued throughout hybrids:

- Let v^j be the random variable that represents the output of the jth experiment (including the view of the adversary and the output of the honest players). To prove security of the MPC, we need to show that the the random variables v^0 and v^6 are computationally indistinguishable.

- Let $\{W_{\mathrm{k}\to\mathrm{i}}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A},P_\mathrm{i}\in\mathcal{H}}$ be the random variables representing the XOR of the values committed in the non-malleable commitment within the NMZK and the mask sent in the fourth round of the NMZK (i.e. $W_{\mathrm{k}\to\mathrm{i}} := \widehat{s}^0_{\mathrm{k}\to\mathrm{i}} \oplus \widehat{s}^1_{\mathrm{k}\to\mathrm{i}}$).

Public-coin property of non-malleable commitment. During our proofs, we reduce our indistinguishability argument to a specific cryptographic property that holds in the stand-alone setting. We might require the non-malleable commitment to interact with an external party R. Note that the simulator will often rewind the adversary. But since R is a stand-alone receiver, its responses can be used only in a single thread.

To deal with this, we do the following. On the main thread, any message from the adversary is forwarded externally to R, and responses from R are forwarded internally to the adversary. But in the look ahead threads, we use the public coin property of the non-malleable commitment to create responses on our own and forward them internally to the adversary.

$\mathbf{H_0}$**:** Execution of the protocol Π in the real world with adversary $\mathcal{A}$.

Soundness lemma. We claim an important lemma that is relevant to the real execution. The lemma says that the adversary $\mathcal{A}$ does not commits to a trapdoor witnesses in the extractable non-malleable commitment inside the non-malleable zero knowledge proofs, where it acts as the prover, if the proof verify. Specifically, as described in Appendix A.5.1, we refer to the XOR of the value inside the non-malleable commitment and the mask sent in the fourth round of the NMZK protocol.

Lemma 2. *Let* $\{\pi_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}},P_{\mathrm{i}}\in\mathcal{H}}$ *be the NMZK proofs for* L *that* $\mathcal{A}$ *sends to all the honest players. Let* $p_{\mathrm{k}\to\mathrm{i}}$ *correspond to the probability that* $\mathrm{W}^0_{\mathrm{k}\to\mathrm{i}}$ *are trapdoor witnesses for the statements being proved in the NMZK above. For the real execution, if all the proofs are accepting, then*

$$p_{\mathrm{k}\to\mathrm{i}} < \nu(n) \qquad \forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{H}$$

for some negligible function ν*.*

Proof. On a high level, the proof follows from the unforgability of the signature scheme and the extractability of the non-malleable commitment.

Without loss of generality assume that the above condition is false. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $p_{\mathrm{k}\to\mathrm{i}}$ is non-negligible.

We will arrive at a contradiction by extracting the masked trapdoor witness from the non-malleable commitment $\pi_{\mathsf{nmcom}_{\mathrm{k}\to\mathrm{i}}}$ to break the underlying signature scheme in the NMZK.

For the proof $\pi_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$, all messages other than the ones for the signature scheme are generated honestly. Messages for the signature scheme are obtained from the external challenger. Note that the last message from the external challenger is in the third round when the challenger signs the message sent to the challenger in the second round. The extraction from the non-malleable commitment is performed by rewinding the third and fourth rounds. Thus, while rewinding we repeatedly send to the adversary the same signature received from the challenger. Since the trapdoor condition consists of two distinct messages with their corresponding signature, from the extracted values and the mask for the non-malleable commitment we have these trapdoor values with non-negligible probability. Since we've queried the external challenger only once, we have broken the underlying signature scheme with a forged signature. □

Consequence of the soundness lemma. If this above property, which will be referred to as the *soundness condition*, holds, then the "trapdoor condition" is false for both the NMZK and WIPoK proofs. We shall maintain this soundness invariant throughout our proof across hybrids by arguing that the distribution of W doesn't change. This means, from the soundness of sWIAoK (used inside NMZK) and WIPoK, the "honest statements" are true if the proofs are accepting.

$\mathbf{H_1}$: Identical to $\mathbf{H_0}$ except that we rewind polynomial number of times in the second and third round to extract the trapdoors for the non-malleable zero knowledge proofs, and the committed values (input and randomness) from the extractable commitment scheme.

We abort with output $\bot_{\mathsf{nmzk}}$ for the hybrid if the extractor for NMZK fails, and abort with output $\bot_{\mathsf{ecom}}$ if the extractor for the commitment fails to return a value.

Since the only difference from the previous hybrid $\mathbf{H_0}$ (real execution) is the rewinding to perform the required extractions, the main thread in the experiment remains unchanged. Thus, we claim the following

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{0}_{\mathrm{k}\to\mathrm{i}} \approx_s W^{1}_{\mathrm{k}\to\mathrm{i}} \tag{1}$$

Let us assume the claim isn't true. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $W^{0}_{\mathrm{k}\to\mathrm{i}}$ and $W^{1}_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by an unbounded adversary D. We use D to create another unbounded adversary D' that distinguishes between the main threads of $\mathbf{H_0}$ and $\mathbf{H_1}$. It works by extracting the commitment in the extractable non-malleable commitment within the NMZK proof $\pi_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}$, and uses D to distinguish between the commitments. Since the main thread is unchanged, this is a contradiction.

We now claim,

$$v^0 \approx_s v^1. \tag{2}$$

If the proofs are accepting in the main thread, then the extracted values (from the extractable commitment) are correct by the soundness lemma and equation 1.

For the view, since the main thread remains the same, all that is left to argue is that the experiment aborts with negligible probability.

From the NMZK we know that the probability $\bot_{\mathsf{nmzk}}$ is output is negligible in n. Similarly, we output $\bot_{\mathsf{ecom}}$ with probability negligible in n. This implies that the experiment aborts with negligible probability, proving the claim.

$\mathbf{H_2}$: Identical to $\mathbf{H_1}$ except that we set the XOR of the value inside the non-malleable commitment $\tau_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$ and the mask $\widehat{s}^{1}_{\mathrm{i}\to\mathrm{k}}$ to be the extracted trapdoor. This is done in each instance of NMZK proofs sent by the honest players and for proofs between honest players too. This is achieved by setting the mask accordingly in the fourth round.

We now claim the following,

$$v^1 \approx_c v^2 \tag{3}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{1}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{2}_{\mathrm{k}\to\mathrm{i}} \tag{4}$$

Equation 4 follows from the non-malleability of the non-malleable commitment $\widehat{\pi}_{\mathsf{nmcom}}$. Specifically, we go from a hybrid where all the masked commitments (XOR of the value inside the non-malleable commitment $\widehat{\pi}_{\mathsf{nmcom}_{\mathrm{i}\to\mathrm{k}}}$ with the mask $\widehat{s}^{1}_{\mathrm{i}\to\mathrm{k}}$ sent in the 4th round) are 0 to a hybrid where all the masked commitments are to the respective trapdoors. To rely on the hiding property of the commitment scheme, we construct intermediate hybrids where in each hybrid we change the value only in one commitment scheme. If equation 4 does not hold, then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $W^{1}_{\mathrm{k}\to\mathrm{i}}$ and $W^{2}_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT adversary D. To reduce to the non-malleability, we need to expose the non-malleable commitment in the protocol to an external challenger and receiver.

For equation 3, we needed to expose it only to the external challenger, and this is done identically here as well.

To do this, we expose a part of the protocol to an external committer C for the non-malleable commitment (challenger). All messages of the protocol other than the ones that differ in the pair of adjacent hybrids is computed as in the previous hybrid. The messages for the specific non-malleable commitment are taken from the external challenger. Responses intended for the challenger from the adversary are forwarded to this challenger. As challenge messages, we send prior to the second round r and $r \oplus \mathsf{td}$ where r is picked randomly, and td is the trapdoor. But the trapdoor is only available after the third round, so we start look ahead threads after the first round to obtain td. This is possible since the verification key vk is fixed in the first round and messages for the identified non-malleable commitment are generated locally in the look up thread. On obtaining the requisite value from the look ahead thread, we continue the main thread from the first round by taking in the non-malleable commitment messages externally. For the mask we send r. Thus if the external challenger committed to r, we are in the first hybrid else we are in the second hybrid.

Lastly, we note that rewinding the second and third rounds are not an issue since the second message of the non-malleable commitment comes from the external challenger and hence just repeated while rewinding. And since the look ahead threads are cut off after the third round, we don't need to worry about the fourth round from the external challenger.

The messages for the non-malleable commitment $\tau_{\mathsf{nmcom}_{\mathrm{k}\to\mathrm{i}}}$ are forwarded to the external receiver and responses back to $\mathcal{A}$. Depending on the value committed by the external challenger, we are in either of the two adjacent hybrids. If the adversary is able to commit to different values in these cases, then it has broken the non-malleability of the non-malleable commitment scheme. Thus equation 4 holds. We use the public-coin property of the non-malleable commitment discussed earlier to respond to messages locally in the look-ahead threads.

Equation 3 follows from the fact the hiding property of the non-malleable commitment.

If equation 3 does not hold, then there are adjacent intermediate hybrids such that they are distinguishable. i.e. when we change the value of the mask $\widehat{s}^1_{\mathrm{i}\to\mathrm{k}}$ such that the XOR with the value inside the non-malleable commitment becomes the trapdoor witness, there is PPT distinguisher D that can differentiate the two cases. We shall use this PPT distinguisher to break the hiding property of the non-malleable commitment. To do this, we expose a part of the protocol to an external committer as described for equation 4.

Now we use the adversary $\mathcal{A}$ that distinguishes the two views to directly distinguish between the values committed to by the external challenger C.

$\mathbf{H_3}$: Identical to $\mathbf{H_2}$ except that we simulate the proofs for both the WIPoK and the NMZK.

We now claim the following,

$$v^2 \approx_c v^3 \tag{5}$$

$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^2_{\mathrm{k}\to\mathrm{i}} \approx_c W^3_{\mathrm{k}\to\mathrm{i}} \tag{6}$$

We argue by splitting into two sub-hybrids:

- First we simulate all the proofs in the WIPoK. This is done by changing the last round of the WIPoK, and hence the last round of the protocol.

$$v^2 \approx_c \widetilde{v}^2 \tag{7}$$

$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^2_{\mathrm{k}\to\mathrm{i}} \approx_c \widetilde{W}^2_{\mathrm{k}\to\mathrm{i}} \tag{8}$$

Equation 8 follows trivially from the fact that the change is made in the fifth round of the protocol after completion of the non-malleable commitment protocol.

For equation 7, this follows directly from the witness indistinguishable property of the WIPoK. Specifically, if adversary $\mathcal{A}$ is able to distinguish between the views, then we can construct an adversary that breaks the witness indistinguishability of the WIPoK. The change in the proofs are made through a sequence of hybrids where in each hybrid only one proof is changed. If the views are distinguishable, there exists an adjacent pair of hybrids such that the change is distinguishable. Let this proof be $\pi_{\mathsf{WIPoK}_{\mathrm{i}\to\mathrm{k}}}$. To reduce the security to the witness indistinguishability, we take this proof from an external challenger. The rewinding for extracting the trapdoors and inputs are only in the second and third round, and hence do not affect the interaction with the external challenger. The external challenger gives a proof using one of the two witnesses, and depending on the witness used we are in either of the two adjacent hybrids. Thus, we use adversary $\mathcal{A}$ that distinguishes the two views to distinguish which witness was used.

– In the hybrid, we simulate all the statistical WIAoK. This the is done by changing the last round of the sWIAoK, and hence the fourth round of the overall protocol.

$$\widetilde{v}^2 \approx_s v^3 \tag{9}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widetilde{W}^2_{\mathrm{k}\to\mathrm{i}} \approx_s W^3_{\mathrm{k}\to\mathrm{i}} \tag{10}$$

Equation 10 follows trivially from the fact that the change made is statistical. Specifically, assume equation 10 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $\widetilde{W}^2_{\mathrm{k}\to\mathrm{i}}$ and $W^3_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by an unbounded adversary D. We use D to create another unbounded adversary D' that breaks the statistical witness indistinguishability. The change in the proofs are made through a sequence of hybrids where in each hybrid only one proof is changed. If the views are distinguishable, there exists an adjacent pair of hybrids such that the change is distinguishable. To reduce the security to the witness indistinguishability, we take this proof from an external challenger. The rewinding for extracting the trapdoors and inputs are only in the second and third round, and hence do not affect the interaction with the external challenger. The external challenger gives a proof using one of the two witnesses, and depending on the witness used we are in either of the two adjacent hybrids.

Next we extract $\widehat{s}^0_{\mathrm{k}\to\mathrm{i}} \oplus \widehat{s}^1_{\mathrm{k}\to\mathrm{i}}$ and use D to distinguish between the two cases, i.e. two witnesses. For equation 9, we set up the experiment to the external challenger exactly as described above. Instead of extracting the non-malleable commitment, we use the adversary $\mathcal{A}$ that distinguishes the two views to directly distinguish between the witnesses used.

$\mathbf{H_4}$: Identical to $\mathbf{H_3}$ except for the following. With $\vec{m}^1, \vec{m}^2, \vec{m}^3, \{(x_k, r_k)\}_{P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}}$, we check if the adversary has followed the computation in the first three rounds correctly. If not, and the proofs for L verify, we output $\bot^1_{\mathsf{rMPC}}$ and abort. If the proofs for L do not verify, send `abort` to the ideal functionality.

Otherwise send the inputs extracted to the ideal functionality to obtain the output. Given the extracted inputs, randomnesses and output we run $\mathsf{Sim}^2_{\mathsf{rMPC}}$ to obtain m^4_i for every honest player P_{i}. Since it is a semi-honest simulator, it simulates the transcript, and we thus also have $\{m^4_k\}_{P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}}$. On receiving $\{m^{*4}_k\}_{P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}}$ from the adversary, checks if the simulated transcript matches the on received. If it does not match, but the WIPoK proofs for L_{WIPoK} verify, output $\bot^2_{\mathsf{rMPC}}$ and abort. If the proofs for L_{WIPoK} do not verify, send `abort` to the ideal functionality.

Conditioned on the fact that we don't abort, we claim the following

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^3_{\mathrm{k}\to\mathrm{i}} \approx_s W^4_{\mathrm{k}\to\mathrm{i}}. \tag{11}$$

Equation 11 is trivially true since the last message that was simulated for the robust semi honest MPC was sent in the fifth round, after the completion of the NMZK for L, and hence after completion of the non-malleable commitment $\tau_{\mathsf{nmcom}_{\mathrm{k}\to\mathrm{i}}}$. Thus the execution thread till the fifth round is statistically indistinguishable.

Lastly, we claim

$$v^3 \approx_c v^4. \tag{12}$$

Before we can reduce this to the security of the underlying simulator for the robust semi honest MPC, we must ensure that the security holds. This is the case when the adversary behaved semi-honestly in the first three rounds, i.e. if the proofs for L verify and the hybrid does not output $\bot^1_{\mathsf{rMPC}}$. From equation 11 and the soundness condition of the previous hybrid, if the proofs for L verify, the adversary does not behave honestly in the first three rounds with only negligible probability. Thus $\bot^1_{\mathsf{rMPC}}$ is output with only negligible probability.

Then, if the above claim does not hold, we break the security of $\mathsf{Sim}^2_{\mathsf{rMPC}}$ (implied by the definition of the robust semi honest MPC). Additionally, if the proofs for L_{WIPoK} verify, from the soundness condition $\bot^2_{\mathsf{rMPC}}$ is output with negligible probability. This ensures the that the output distribution of the honest parties is indistinguishable from the previous hybrid. This proves our claim.

$\mathbf{H_5}$**:** Identical to $\mathbf{H_4}$ except we commit to the '0' string in the extractable commitment. This applies to commitments within every pair of honest players as well.

We do this through two sub-hybrids.

$\mathbf{H_{5,0}}$: In this sub-hybrid, we "cheat" in the third round everywhere by sending random strings as response to the challenges in the look ahead threads. The changes here are only statistical. Thus we have,

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \quad `W^4_{\mathrm{k}\to\mathrm{i}} \approx_s W^{5,0}_{\mathrm{k}\to\mathrm{i}} \tag{13}$$

$$v^4 \approx_s v^{5,0}. \tag{14}$$

$\mathbf{H_5}$: We commit to the '0' string in the extractable commitments. We're able to do this because the decommitment information is not used anywhere else.

We claim the following,

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{5,0}_{\mathrm{k}\to\mathrm{i}} \approx_c W^5_{\mathrm{k}\to\mathrm{i}}. \tag{15}$$

Let us assume the claim isn't true. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $W^{5,0}_{\mathrm{k}\to\mathrm{i}}$ and $W^5_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT adversary D. We use D to create another PPT adversary D' that breaks the hiding property of the extractable commitment. Specifically, we do this by a sequence of intermediate hybrids where we change the value of only one commitment at a time. If the claim isn't true, then there exists adjacent intermediate hybrids such that they are distinguishable. Let the intermediate hybrids be such that only the commitment $r^1_{\tau_{\mathrm{i}\to\mathrm{k}}}$ was changed. For a reduction to the hiding property of the extractable commitment we need to expose this commitment to an external challenger. i.e. all messages apart from the commitment $\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ are computed as in the previous hybrid. The messages $\tau_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ are taken from the external challenger C and the subsequent response by the adversary is forwarded to the challenger and so on. Recall that the adversary is being rewound in the second and third round in order to extract the inputs and trapdoors, but from the previous sub-hybrid, we're answering them randomly on the look ahead threads (this change is only statistical). As challenge messages, prior to the first round we send 0 and $(x_{\mathrm{i}}, r_{\mathrm{i}})$. Thus, depending on the value committed to by the challenger, we are in one of the two adjacent hybrids.

Now we rewind the adversary in the third and fourth round to extract the non-malleable commitment to obtain $\widehat{s}^0_{\mathrm{k}\to\mathrm{i}} \oplus \widehat{s}^1_{\mathrm{k}\to\mathrm{i}}$. Only the last message of the extractable commitment from the external challenger C overlaps with the third and fourth round and hence resent unchanged during the extraction. We now present the extracted value to the PPT D to distinguish between the two hybrids and thus breaking the hiding of the extractable commitment scheme.

We claim the following

$$v^{5,0} \approx_c v^5. \tag{16}$$

Equation 16 follows from the hiding property and rewinding security of the extractable commitment scheme. We set up the experiment to the external challenger exactly as described above. Instead of extracting the non-malleable commitment, we use the adversary $\mathcal{A}$ that distinguishes the two views to directly distinguish between the values committed to by the external challenger.

$\mathbf{H_6}$: Identical to $\mathbf{H_5}$ except that we use $\mathsf{Sim}^1_{\mathsf{rMPC}}$ to simulate the first three round of the honest players in Π_{rMPC}.

We claim the following,

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^5_{\mathrm{k}\to\mathrm{i}} \approx_c W^6_{\mathrm{k}\to\mathrm{i}}. \tag{17}$$

Equation 17 follows from the computational indistinguishability of the the view output by $\mathsf{Sim}^1_{\mathsf{rMPC}}$ from the real view implicit from the definition of the robust semi honest MPC. This is where the "special rewinding" property (see B.6) is used. Let us assume the claim isn't true. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $W^5_{\mathrm{k}\to\mathrm{i}}$ and $W^6_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT adversary D. We use D to create another PPT adversary D' that breaks the security of $\mathsf{Sim}^1_{\mathsf{rMPC}}$. We take the transcript of the first three rounds of the robust MPC externally and force it on the adversary (by setting random coins accordingly). Recollect that we're still extracting the input and trapdoor by rewinding in the second and third round. Here the "special rewinding" property is used. In the look ahead threads, if we send a different message in the second round while rewinding, the adversary might send a different second message of the robust semi-honest MPC. To be able to extract from the non-malleable commitment, we need to complete the fourth round of the protocol, and hence need the "special rewinding" property to simulate messages for potentially different second round messages in the look ahead threads.

Now we rewind the adversary in the third and fourth round to extract the non-malleable commitment to obtain $\widehat{s}^0_{\mathrm{k}\to\mathrm{i}} \oplus \widehat{s}^1_{\mathrm{k}\to\mathrm{i}}$. Only one round of the robust MPC overlaps with the rewinding rounds for the extraction, and we send the same message (received from the challenger) in each look ahead thread. We now present the extracted value to the PPT D to distinguish between the two hybrids and thus breaking the hiding of the extractable commitment scheme.

The extraction probability does not change because of indistinguishability of the view in each of the adversary's (look ahead) threads (from the special rewinding property). One can alternatively considering an intermediate hybrid where the changes are initially made only in the look-ahead threads and then the changes on the main thread.

Lastly, we claim the following

$$v^5 \approx_c v^6. \tag{18}$$

Equation 18 follows from the computational indistinguishability of the the view output by $\mathsf{Sim}^1_{\mathsf{rMPC}}$ from the real view implicit from the definition of the robust semi honest MPC. We set up the experiment to the external challenger exactly as described above for proving equation 17. Instead of extracting the non-malleable commitment, we use the adversary $\mathcal{A}$ that distinguishes the two views to directly distinguish between the transcripts sent by the challenger.

Hybrid $\mathbf{H_6}$ is identical to our simulator. From the above discussion, we have

$$v^0 \approx_c v^6$$

thus proving security of the constructed MPC.

D Proof of Theorem 10

We present the proof for our four round construction below. As before, we discuss a few properties of the underlying primitives that we will need:

- The simulator for the robust semi honest MPC, as previously discussed, consists of two parts. The first part, $\mathsf{Sim}^1_{\mathsf{rMPC}}$, simulates the first three rounds of the robust semi honest MPC without requiring inputs or outputs of the adversary. The second part, $\mathsf{Sim}^2_{\mathsf{rMPC}}$, when given the inputs and outputs of the adversary simulates the last message of robust semi honest MPC. Additionally, note that this simulation works only in the semi-honest setting.
- The extractor for the 3 round "rewinding secure" extractable commitment works by rewinding the second and third round polynomial number of times. From the Lemma 1, we know that if the commitments are well formed, extraction of the correct inputs fail with only negligible probability.
- The simulator of the NMZKs works by extracting a trapdoor. Specifically, it rewinds the second and third round polynomial number of times to get signatures for two distinct messages. Further, this extraction fails only with negligible probability if the adversary does not abort with non-negligible probability.
- Combining the above two properties, we see that the rewindings of NMZK and the extractable commitment are "composable" because they rewind in the same rounds in our MPC protocol.
- To extract the value in the four round non-malleable commitment within the NMZK, we rewind in the third and fourth round. This will be useful will arguing the proofs in the hybrids.

D.1 Description of the Simulator

We describe the ideal world simulator Sim below.

1. The first three rounds of protocol are simulated by picking random inputs for the honest parties and behaving "honestly" with these inputs as follows:
   - For the robust semi honest MPC, pick random $\{x'_{\mathsf{i}}, r'_{\mathsf{i}}\}_{P_{\mathsf{i}} \in \mathcal{H}}$ as inputs to the first three rounds. We shall use the last round of the robust semi honest MPC to correct the output. We obtain $\{m^1_{\mathsf{i}}, m^2_{\mathsf{i}}, m^3_{\mathsf{i}}\}_{P_{\mathsf{i}} \in \mathcal{H}}$.
   - For the extractable commitment, we deal with two cases:
     (a) For commitments from the honest players to the adversary, we commit honestly to random strings $\{r^0_{\mathsf{ecom}_{\mathsf{i}\to\mathsf{k}}}\}_{P_{\mathsf{i}} \in \mathcal{H}}$. We do this for commitments within the honest players as well. In the third round, we send $\{r^1_{\mathsf{ecom}_{\mathsf{i}\to\mathsf{k}}} := (x'_{\mathsf{i}}, r'_{\mathsf{i}}) \oplus r^0_{\mathsf{ecom}_{\mathsf{i}\to\mathsf{k}}}\}_{P_{\mathsf{i}} \in \mathcal{H}}$ where $x'_{\mathsf{i}}, r'_{\mathsf{i}}$ are the inputs and randomness generated in the previous step.
     (b) For commitments where the honest players are recipients, run the extractor to send both the responses, and extract the values inside the commitments. If the extraction fails to return a value, output $\perp_{\mathsf{ecom}}$ and abort.

   This gives us $\left\{\pi^{\mathsf{j}}_{\mathsf{ecom}_{\mathsf{i}}}\right\}_{\mathsf{j} \in \{1,2,3\}, P_{\mathsf{i}} \in \mathcal{H}}$, $\{r^1_{\mathsf{ecom}_{\mathsf{i}\to\mathsf{k}}}\}_{P_{\mathsf{i}} \in \mathcal{H}}$ and the extracted commitments.

- We generate $\{y_{\mathrm{i}}\}_{P_{\mathrm{i}}\in\mathcal{H}}$ honestly as defined by the protocol.
- For the input delayed WIPoK, we behave honestly with our randomly generated inputs and randomness. This gives us $\left\{\pi^{\mathrm{j}}_{\mathsf{WIPoK_i}}\right\}_{\mathrm{j}\in\{1,2,3\},P_{\mathrm{i}}\in\mathcal{H}}$. If any of the proofs received at the end of the third round fails, output $\perp$ and abort.
- For simulating proofs for the NMZKs, we deal with three different cases:
  (a) For proofs from the adversary, the honest player acts as a verifier. In this case, fix a random tape for the verifier and respond honestly to adversary queries.
  (b) For proofs within honest players, we fix the random tape for the verifiers and thus can trivially compute the trapdoor in the NMZKs for both languages using the verifier's random tape.
  (c) For proofs from honest players to the adversary, we run the simulator $\mathsf{Sim_{nmzk}}$. This internally rewinds polynomial many times to obtain the trapdoors. If the extractor fails, output $\perp_{\mathsf{nmzk}}$ and abort.

  This gives us $\left\{\pi^{\mathrm{j}}_{\mathsf{nmzk_i}}\right\}_{\mathrm{j}\in\{1,2,3\},P_{\mathrm{i}}\in\mathcal{H}}$ and the extracted trapdoors.

  As noted earlier, the rewinding performed within the NMZK simulator and the extractor for extractable commitments work in the same rounds and can be done for each without affecting the other.
- For the non-malleable commitment, as above, we deal with two cases:
  (a) For commitments from the honest players to the adversary, we commit to 0. For the for non-malleable commitments within the NMZK as well we commit to the trapdoors (extracted earlier) of the NMZK. By the construction of the NMZK protocol in [COSV17], the change for this is made only in the fourth round where we send a different mask s_1. For the two round non-malleable commitment, the committed value is set in the second round.
  (b) For commitments where the honest players are recipients, we behave honestly.

2. The last round is simulated as below:

- Send the extracted inputs $\{x_k\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}}}$ to the ideal functionality to obtain the output y. Obtain the final round of the robust semi honest MPC as

$$\{m^4_{\mathrm{i}}\}_{P_{\mathrm{i}}\in\mathcal{P}} \leftarrow \mathsf{Sim}^2_{\mathsf{rMPC}}\left(\vec{m}^1,\vec{m}^2,\vec{m}^3,\{x_{\mathrm{k}}\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}}},\{r_{\mathrm{k}}\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}}},y\right).$$

- Simulate the last round of the NMZK for L in two steps.
  - For proofs from the honest parties to the adversary, use $\mathsf{Sim_{nmzk}}$ with inputs:

$$\left\{\pi^{\mathrm{j}}_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{j}\in\{1,2,3\},\ P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}},P_{\mathrm{i}}\in\mathcal{H}}$$

    and the trapdoors obtained earlier to compute

$$\left\{\pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}\right\}_{P_{\mathrm{k}}\ \in\mathcal{P}^{\mathcal{A}},\ P_{\mathrm{i}}\ \in\mathcal{H}}.$$

  - For proofs within honest parties, the trapdoor is trivially known to the adversary and thus use $\left\{\pi^{\mathrm{j}}_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}\right\}_{\mathrm{j}\in\{1,2,3\},\ P_{\mathrm{k}},P_{\mathrm{i}}\in\mathcal{H}}$ to construct

$$\left\{\pi^4_{\mathsf{nmzk}_{\mathrm{i}\to\mathrm{k}}}\right\}_{P_{\mathrm{k}},\ P_{\mathrm{i}}\ \in\mathcal{H}}.$$

This gives us the required $\left\{\pi^4_{\mathsf{nmzk}_\mathrm{i}}\right\}_{P_\mathrm{i}\ \in\mathcal{H}}$.

On receiving the proofs from the adversary check if all the received proofs are valid i.e. verify if $\{\pi^\mathrm{j}_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}\}_{\mathrm{j}\in\{1,2,3,4\},\ P_\mathrm{k}\ \in\mathcal{P}^\mathcal{A},\ P_\mathrm{i}\ \in\mathcal{H}}$ are valid proofs in L. If the check fails, send `abort` to the ideal functionality. if the proofs verify, but $\{m^{*4}_\mathrm{k}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A}}$ differs from the simulated transcript, output $\bot^2_{\mathsf{rMPC}}$.

The simulator behaves honestly with respect to the WIPoK and doesn't use the trapdoor. But in the hybrids, we shall use a hybrid simulator that will prove the trapdoor statement in the WIPoK.

D.2 Description of the hybrids

Random variables. As in the five round setting, we introduce the following random variables and their indistinguishability will be argued throughout hybrids:

- Let v^j be the random variable that represents the output of the jth experiment (including the view of the adversary and the output of the honest players).
- Let $\{W_{\mathrm{k}\to\mathrm{i}}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A},P_\mathrm{i}\in\mathcal{H}}$ be the random variables representing the values that are committed in the non-malleable commitments of Π_{nmcom}. On the other hand, $\left\{\widehat{W}_{\mathrm{k}\to\mathrm{i}}\right\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A},P_\mathrm{i}\in\mathcal{H}}$ are the random variables representing the XOR of the values committed in the non-malleable commitment within the NMZK and the mask sent in the fourth round of the NMZK (i.e. $\widehat{W}_{\mathrm{k}\to\mathrm{i}} := s^0_{\mathrm{k}\to\mathrm{i}} \oplus s^1_{\mathrm{k}\to\mathrm{i}}$). We need these to ensure that adversary behaves in a semi-honest way for the computation of the robust semi honest MPC.

$\mathbf{H_0}$: Execution of the protocol Π in the real world with adversary $\mathcal{A}$.

Soundness lemma.

Lemma 3. *Let* $\{\pi_{\mathsf{WIPoK}_{\mathrm{k}\to\mathrm{i}}}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A},P_\mathrm{i}\in\mathcal{H}}$ *and* $\{\pi_{\mathsf{nmzk}_{\mathrm{k}\to\mathrm{i}}}\}_{P_\mathrm{k}\in\mathcal{P}^\mathcal{A},P_\mathrm{i}\in\mathcal{H}}$ *be the input delayed* WIPoK *proofs for* L_{WIPoK} *and NMZK proofs for* $\widehat{L}$ *respectively that* $\mathcal{A}$ *sends to the honest players. Let* $p_{\mathrm{k}\to\mathrm{i}}$ *correspond to the probability that* $W^0_{\mathrm{k}\to\mathrm{i}}$ *is the trapdoor witness for the statement in* L *being proved in the input delayed* WIPoK *above. Similarly,* $\widehat{p}_{\mathrm{k}\to\mathrm{i}}$ *corresponds to the probability that* $\widehat{W}^0_{\mathrm{k}\to\mathrm{i}}$ *is the trapdoor witness for* $\widehat{L}$. *For the real execution, if the proofs are accepting, then*

$$p_{\mathrm{k}\to\mathrm{i}}, \widehat{p}_{\mathrm{k}\to\mathrm{i}} < \nu(n) \qquad \forall P_\mathrm{k}\in\mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i}\in\mathcal{H}$$

for some negligible function ν.

Proof. The high level idea of the proof is the following: Suppose by contradiction the lemma is not true, then there $\exists P_\mathrm{k}\in\mathcal{P}^\mathcal{A}, P_\mathrm{k}\in\mathcal{H}$ such that either $p_{\mathrm{k}\to\mathrm{i}}$ or $\widehat{p}_{\mathrm{k}\to\mathrm{i}}$ is non-negligible.

Consider the case that $p_{\mathrm{k}\to\mathrm{i}}$ is non-negligible. Then we shall construct an adversary $\mathcal{A}_{\mathsf{Sign}}$ that breaks the signature scheme used in the underlying NMZK. We shall do this by extracting from the non-malleable commitment within the NMZK. Recall that this non-malleable commitment is a 4 round protocol whose values can be extracted by rewinding the third and fourth round messages. For P_i receiving a proof, all messages, other than the ones relevant to the signature scheme, are sent honestly. Specifically, in the first round, the verification key vk sent by the challenger for Sign is forwarded to the adversary. When the adversary sends the message to be signed, it is forwarded to the challenger, and the response (signature) forwarded to the adversary. Now we extract from the 4 round non-malleable commitment. This is done by rewinding the third and fourth round messages of the protocol. Only the signature from the challenger overlaps with the round that are rewound. Thus, we can resend the same message to the adversary while rewinding without having to make another query to the challenger. Thus, with non-negligible probability we extract the trapdoor witness for the statement (signature of two distinct messages) and break the signature scheme.

Now, consider the case that $\widehat{p}_{\mathrm{k}\to\mathrm{i}}$ is non-negligible. Here we shall construct an adversary $\mathcal{A}_f$ that breaks the one-wayness of f. For P_{i} receiving a proof, all messages, other than $y_{\mathrm{k}\to\mathrm{i}}$, are sent honestly. We forward the challenge received from the challenger for f. We now extract from the 2 round non-malleable commitment. Since the only message sent by the challenger is in the first round, the rewinding is done completely independent of the challenger interaction. Thus, with non-negligible probability we extract the trapdoor witness for the statement (the pre-image) and break the one-wayness of f. Here we require $T_f >> \widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$.

Thus from the security of the one-way permutation and the signature scheme, the lemma holds. □

Consequence of the soundness lemma. If this above property, which will be referred to as the *soundness condition*, holds, then the "trapdoor condition" is false for both the NMZK and WIPoK proofs. We shall maintain this soundness invariant throughout our proof across hybrids by arguing that the distribution of W doesn't change. This means, from the soundness of sWIAoK (used inside NMZK) and WIPoK, the "honest statements" are true if the proofs are accepting.

$\mathbf{H_1}$: Identical to $\mathbf{H_0}$ except that we rewind polynomial number of times in the second and third round to extract the trapdoors for the NMZK proofs, and the committed values (input and randomness) from the extractable commitment scheme.

We abort with output $\bot_{\mathsf{nmzk}}$ for the hybrid if the extractor for NMZK fails, and abort with output $\bot_{\mathsf{ecom}}$ if the extractor for the extractable commitment fails to return any output. At this point we do not know if the extracted values are indeed the adversary's input and randomness.

Since the only difference from the previous hybrid $\mathbf{H_0}$ (real execution) is the rewinding to perform the required extractions, the main thread in the experiment remains unchanged. Thus, we claim the following

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^0_{\mathrm{k}\to\mathrm{i}} \approx_s W^1_{\mathrm{k}\to\mathrm{i}} \tag{19}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^0_{\mathrm{k}\to\mathrm{i}} \approx_s \widehat{W}^1_{\mathrm{k}\to\mathrm{i}} \tag{20}$$

The proofs follows exactly the same way as in the five round case.

We now claim,

$$v^0 \approx_s v^1. \tag{21}$$

Since the main thread remains the same, all that is left to argue is that the experiment aborts with negligible probability.

From the property of the NMZK and extractable commitments, we know that the probability $\bot_{\mathsf{nmzk}}$ or $\bot_{\mathsf{ecom}}$ is output is negligible in the security parameter. This implies that the experiment aborts with negligible probability, proving the claim.

$\mathbf{H_2}$: Identical to $\mathbf{H_1}$ except that we commit to the trapdoor values in the non-malleable commitments inside the NMZKs sent by the honest players. This is done by changing the mask sent in the fourth message (within the NMZK).

We claim the following,

$$v^1 \approx_c v^2 \tag{22}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^1_{\mathrm{k}\to\mathrm{i}} \approx_s W^2_{\mathrm{k}\to\mathrm{i}} \tag{23}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^1_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^2_{\mathrm{k}\to\mathrm{i}}. \tag{24}$$

Equation 23 trivially follows from the fact that the change is made after the completion of the two round non-malleable commitment.

The proofs for equations 22 and 23 follow identically from the proof of in hybrid $\mathbf{H_2}$ in the five round case.

$\mathbf{H_3}$**:** Identical to $\mathbf{H_2}$ except that we use the trapdoors obtained earlier to simulate the last message of the sWIAoK, from the honest players to the adversarial players, within the NMZK for language $\widehat{L}$. For proofs between any two honest players, since the simulator controls both players, it fixes the random tapes used for the NMZK and thus knowns the trapdoors. Thus, proofs between honest parties are trivially simulated. For proofs from the adversary, we respond honestly.

We now claim the following,

$$v^2 \approx_s v^3 \tag{25}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^2_{\mathrm{k}\to\mathrm{i}} \approx_s W^3_{\mathrm{k}\to\mathrm{i}} \tag{26}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^2_{\mathrm{k}\to\mathrm{i}} \approx_s \widehat{W}^3_{\mathrm{k}\to\mathrm{i}} \tag{27}$$

Equation 26 trivially holds because changes made are in the fourth round after the completion of the two round non-malleable commitment.

Equation 27 follows trivially from the fact that the change made is statistical. Specifically, assume equation 27 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{P}$ such that $\widehat{W}^2_{\mathrm{k}\to\mathrm{i}}$ and $\widehat{W}^3_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by an unbounded adversary D. We use D to create another unbounded adversary D' that breaks the statistical witness indistinguishability. The change in the proofs are made through a sequence of hybrids where in each hybrid only one proof is changed. If the views are distinguishable, there exists an adjacent pair of hybrids such that the change is distinguishable. To reduce the security to the witness indistinguishability, we take this proof from an external challenger. The rewinding for extracting the trapdoors and inputs are only in the second and third round, and hence do not affect the interaction with the external challenger. The external challenger gives a proof using one of the two witnesses, and depending on the witness used we are in either of two adjacent hybrids.

Next we extract $\widehat{s}^0_{\mathrm{k}\to\mathrm{i}} \oplus \widehat{s}^1_{\mathrm{k}\to\mathrm{i}}$ and use D to distinguish between the two cases, i.e. two witnesses.

For equation 25, we set up the experiment to the external challenger exactly as described above. Instead of extracting the non-malleable commitment, we use the adversary $\mathcal{A}$ that distinguishes the two views to directly distinguish between the witnesses used.

$\mathbf{H_4}$**:** Identical to $\mathbf{H_3}$ except for the following changes. Send the inputs extracted to the ideal functionality to obtain the output.

Given the extracted inputs, randomnesses and output, we run $\mathsf{Sim}^2_{\mathsf{rMPC}}$ to simulate m^4_i for every honest player P_{i}.

Since it is a semi-honest simulator, it simulates the transcript, and we thus also have $\left\{m^4_k\right\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}}}$. On receiving $\left\{m^{*4}_k\right\}_{P_{\mathrm{k}}\in\mathcal{P}^{\mathcal{A}}}$ from the adversary, checks if the simulated transcript matches the on received. If it does not match, but the NMZK proofs for $\widehat{L}$ verify, output $\perp^2_{\mathsf{rMPC}}$ and abort. If the proofs for $\widehat{L}$ do not verify, send `abort` to the ideal functionality.

We claim the following

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^3_{\mathrm{k}\to\mathrm{i}} \approx_s W^4_{\mathrm{k}\to\mathrm{i}}. \tag{28}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^3_{\mathrm{k}\to\mathrm{i}} \approx_s \widehat{W}^4_{\mathrm{k}\to\mathrm{i}}. \tag{29}$$

Equations 28 trivially holds because changes made in the hybrid are after the completion of the two round non-malleable commitment.

Equation 29 holds, else we can extract by rewinding to build a distinguisher for $\mathsf{Sim}^2_{\mathsf{rMPC}}$. The proof is identical to the one in the 5 round setting.

Equation 29 holds from the security of the the robust semi honest MPC. Assume equations 29 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $\widehat{W}^3_{\mathrm{k}\rightarrow\mathrm{i}}$ and $\widehat{W}^4_{\mathrm{k}\rightarrow\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the security of the robust semi-honest MPC. To reduce to the security of the robust semi honest MPC, we expose the last round of the robust semi honest MPC to an external challenger. We then rewind to extract the corresponding value from the non-malleable commitment in the NMZK. We then use D to break the security of the robust semi honest MPC. While rewinding the third and fourth round, only the last round is taken externally. In the look ahead threads, for the third round we send the honestly computed third round of the robust semi-honest MPC. From the soundness condition of the WIPoK and equation 28, the proof verifies only if the adversary sent the correct message in the third round for the robust semi honest MPC. Thus in the look ahead threads we can resend the same message obtained from the external challenger in each look ahead thread.

Lastly, we claim

$$v^3 \approx_c v^4. \tag{30}$$

If the proofs for $\widehat{L}$ verify, from the soundness condition $\perp^2_{\mathsf{rMPC}}$ is output with negligible probability. The claim follow from the security of $\mathsf{Sim}^2_{\mathsf{rMPC}}$ implicit from the definition of the robust semi-honest MPC. We set up the experiment exactly like above, but instead of rewinding to extract from the non-malleable commitment, we use $\mathcal{A}$ that distinguishes the views to break the the security of the robust semi honest MPC [10].

Leveled security. We assume the following, and set the security parameters for the primitives accordingly.

- $T_f >> \widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$;
- $T^{\mathsf{h}}_{\mathsf{nmcom}}, T^{\mathsf{nm}}_{\mathsf{nmcom}} >> T_f$;
- $T_{\mathsf{WIPoK}} >> T_f, T_{\mathsf{Sign}}, T_{\mathsf{ecom}}$;
- $T_{\mathsf{rMPC}_{(1-3)}} >> T_f, T_{\mathsf{Sign}}, T_{\mathsf{ecom}}$;
- $T_{\mathsf{ecom}} >> T_f$.

where T_{prim} means that the primitive prim is secure against adversaries running in time T_{prim}, and $T' >> T$ means that $T' > T \cdot \mathsf{poly}(n)$. Specifically $T_{\mathsf{rMPC}_{(1-3)}}$ means that we require the first three rounds of our robust MPC to be indistinguishable (for adversaries running in time $T_{\mathsf{rMPC}_{(1-3)}}$) for any two sets of inputs and randomnesses. In fact, in our construction, the simulator Sim^1 works by setting a random input to generate the first three rounds. Hence, for our construction, we require $T_{\mathsf{rMPC}_{(1-3)}}$-security for the following two distributions: $\mathsf{RealExec}^{\mathcal{A}^1}_{(t-1)}(\vec{x}, z)$ and $\mathsf{Sim}^1(z)$. Further, for the two round non-malleable commitment, we have three parameters $T^{\mathsf{h}}_{\mathsf{nmcom}}$, $T^{\mathsf{nm}}_{\mathsf{nmcom}}$ and $\widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$. $T^{\mathsf{h}}_{\mathsf{nmcom}}$ and $T^{\mathsf{nm}}_{\mathsf{nmcom}}$ indicate that the hiding and non-malleability respectively of the non-malleable commitment hold against adversaries running in time $T^{\mathsf{h}}_{\mathsf{nmcom}}$ and $T^{\mathsf{nm}}_{\mathsf{nmcom}}$. $\widetilde{T}^{\mathsf{ext}}_{\mathsf{nmcom}}$ refers to the running time of the rewinding extractor for the non-malleable commitment.

[10] For our construction, the views are statistically close. This is because changes in the last message are due to reverse computation of output shares.

$\mathbf{H_5}$: Identical to $\mathbf{H_4}$ except that we break the one-way permutation f, to obtain the pre-image ρ which is committed to in the two round non-malleable commitment.

We claim the following

$$v^4 \approx_c v^5. \tag{31}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^4_{\mathrm{k}\rightarrow\mathrm{i}} \approx_c W^5_{\mathrm{k}\rightarrow\mathrm{i}}. \tag{32}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^4_{\mathrm{k}\rightarrow\mathrm{i}} \approx_c \widehat{W}^5_{\mathrm{k}\rightarrow\mathrm{i}}. \tag{33}$$

Assume equations 33 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $\widehat{W}^4_{\mathrm{k}\rightarrow\mathrm{i}}$ and $\widehat{W}^5_{\mathrm{k}\rightarrow\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the hiding property of the input delayed non-malleable commitment. Specifically, we rewind to extract the corresponding non-malleable commitment $\widehat{W}_{\mathrm{k}\rightarrow\mathrm{i}}$. We move from the previous hybrid to the current one by a sequence of intermediate hybrids where in each hybrid we change the value only in one commitment. If equation 33 does not hold, then two adjacent hybrids are distinguishable. Let equation 33 not hold when we change the commitment in $\pi_{\mathsf{nmcom}_{\mathrm{i}^*\rightarrow\mathrm{k}^*}}$. To reduce to the hiding property of the non-malleable commitment, we expose $\pi_{\mathsf{nmcom}_{\mathrm{i}^*\rightarrow\mathrm{k}^*}}$ to an external challenger. All messages other than those of $\pi_{\mathsf{nmcom}_{\mathrm{i}^*\rightarrow\mathrm{k}^*}}$ are computed as in the previous hybrid. Since the commitment is input delayed, we send the challenge messages only in the third round. The challenges sent are 0 and ρ. Depending on the value committed we are in one of the two adjacent hybrids. We then rewind the third and fourth rounds of the non-malleable commitments inside the NMZK to extract the masked value. Since only the third round of the non-malleable commitment (from committer) overlaps with the rewinding, we send the same message in each look-ahead thread. We then use D to break the hiding property of the non-malleable commitment. We require $T^{\mathsf{h}}_{\mathsf{nmcom}} >> T_f$.

Equation 32 holds from the non-malleability of the non-malleable commitment. Assume equations 32 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $W^4_{\mathrm{k}\rightarrow\mathrm{i}}$ and $W^5_{\mathrm{k}\rightarrow\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the non-malleability of the input delayed non-malleable commitment. As before, we move from the previous hybrid to the current one by a sequence of intermediate hybrids where in each hybrid we change the value only in one commitment. If equation 32 does not hold, then two adjacent hybrids are distinguishable. Let equation 32 not hold when we change the commitment in $\pi_{\mathsf{nmcom}_{\mathrm{i}^*\rightarrow\mathrm{k}^*}}$. To reduce to the non-malleability, as explained for 33, we expose the non-malleable commitment to an external committer and additionally to an external receiver. The challenge messages to the external committer are the same as earlier, we forward the commitment messages for $\pi_{\mathsf{nmcom}_{\mathrm{k}\rightarrow\mathrm{i}}}$ to the external receiver. If depending on the value committed by the external committer, the committed value to the external receiver changes, we've broken the non malleability of non-malleable commitment scheme. We require $T^{\mathsf{nm}}_{\mathsf{nmcom}} >> T_f$.

Equation 31 follows from the hiding property of the non-malleable commitment, and the fact that $T^{\mathsf{h}}_{\mathsf{nmcom}} >> T_f$, and the hybrid thus takes time $O(max\{T_f\})$ which is in turn less than $T^{\mathsf{h}}_{\mathsf{nmcom}}$. The experiment is set up exactly as for equation 33, but instead of rewinding to extract from the non-malleable commitment, we use $\mathcal{A}$ that distinguishes the views to break the hiding property of the non-malleable commitment.

$\mathbf{H_6}$: Identical to $\mathbf{H_5}$ except that we stop rewinding to extract the input and trapdoor, and instead break the signature scheme and the extractable commitment to obtain the trapdoors and the adversary's inputs. Specifically, we break the hiding of the extractable commitment scheme and additionally use the mask to get the input, and break the signature scheme to compute two signatures.

We claim the following

$$v^5 \approx_c v^6. \tag{34}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^5_{\mathrm{k}\to\mathrm{i}} \approx_c W^6_{\mathrm{k}\to\mathrm{i}}. \tag{35}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^5_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^6_{\mathrm{k}\to\mathrm{i}}. \tag{36}$$

Equation 34 follows from the fact that this was only a statistical change. This is because the main thread remains unchanged when we stop rewinding to extract by breaking the underlying schemes. For the same reason, equations 35 and 36 and hold. Note that we're still verifying the proofs to validate the extracted values.

$\mathbf{H_7}$: Identical to $\mathbf{H_6}$ except that we use the trapdoor witnesses in the WIPoK sent by the honest parties. This change is made only in the third round of the WIPoK and thus also the third round of the overall protocol.

We claim the following

$$v^6 \approx_c v^7. \tag{37}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^6_{\mathrm{k}\to\mathrm{i}} \approx_s W^7_{\mathrm{k}\to\mathrm{i}}. \tag{38}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^6_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^7_{\mathrm{k}\to\mathrm{i}}. \tag{39}$$

Assume equations 39 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $\widehat{W}^5_{\mathrm{k}\to\mathrm{i}}$ and $\widehat{W}^6_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the witness indistinguishability of the input delayed WIPoK. Specifically, we rewind to extract the corresponding non-malleable commitment $\widehat{W}_{\mathrm{k}\to\mathrm{i}}$. We move from the previous hybrid to the current one by a sequence of intermediate hybrids where in each hybrid we change the witness in a single proof. If equation 39 does not hold, then two adjacent hybrids are distinguishable. Let equation 39 not hold when we change the witness in proof $\pi_{\mathsf{WIPoK}_{\mathrm{i}^*\to\mathrm{k}^*}}$. To reduce to the witness indistinguishability of the WIPoK, we expose $\pi_{\mathsf{WIPoK}_{\mathrm{i}^*\to\mathrm{k}^*}}$ to an external challenger. All messages other than those of $\pi_{\mathsf{WIPoK}_{\mathrm{i}^*\to\mathrm{k}^*}}$ are computed as in the previous hybrid. Depending on the witness used by the external challenger we are in one of the two adjacent hybrids. We then rewind the third and fourth rounds of the non-malleable commitments inside the NMZK to extract the masked value. Since only the third round of the WIPoK (from prover) overlaps with the rewinding, we send the same message in each look-ahead thread. We then use D to break the witness indistinguishability of the WIPoK.

Equation 38 trivially holds from the fact that the changes made in this hybrid are after completion of the two round non-malleable commitment.

Equation 37 follows from the witness indistinguishability property of the input delayed WIPoK, and the fact that $T_{\mathsf{WIPoK}} >> T_f$, $T_{\mathsf{WIPoK}} >> T_{\mathsf{ecom}}$ and $T_{\mathsf{WIPoK}} >> T_{\mathsf{Sign}}$. We need the latter two because we're still breaking the primitives to extract, and the hybrid thus takes time $O(max\{T_f, T_{\mathsf{ecom}}, T_{\mathsf{Sign}}\})$ which is in turn less than T_{WIPoK}. The experiment is set up exactly as for equation 39, but instead of rewinding to extract from the non-malleable commitment, we use $\mathcal{A}$ that distinguishes the views to break the witness indistinguishability.

$\mathbf{H_8}$: Identical to $\mathbf{H_7}$ except that we use randomly generated inputs $\{x'_{\mathrm{i}}, r'_{\mathrm{i}}\}_{P_{\mathrm{i}} \in \mathcal{H}}$ as inputs to the first three rounds of the robust semi honest MPC.

We claim the following

$$v^7 \approx_c v^8. \tag{40}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^7_{\mathrm{k}\to\mathrm{i}} \approx_c W^8_{\mathrm{k}\to\mathrm{i}}. \tag{41}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^7_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^8_{\mathrm{k}\to\mathrm{i}}. \tag{42}$$

Assume equations 42 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $\widehat{W}^{7}_{\mathrm{k}\to\mathrm{i}}$ and $\widehat{W}^{8}_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to distinguish the first three rounds of the robust semi-honest MPC. Specifically, we rewind to extract the corresponding non-malleable commitment $\widehat{W}_{\mathrm{k}\to\mathrm{i}}$. To reduce to security of the first three rounds of the robust semi-honest MPC, we expose it to to an external challenger. All messages other than those of the first three rounds of the robust semi-honest MPC are computed as in the previous hybrid. Depending on the transcript sent by the challenger we are in one of two adjacent hybrids. We force this transcript on the adversary by appropriately setting its random coins. We then rewind the third and fourth rounds of the non-malleable commitments inside the NMZK to extract the masked value. Since only the third round of the robust semi-honest MPC overlaps with the rewinding, we send the same message in each look-ahead thread. We then use D to break the security of the first three rounds of the robust semi-honest MPC. Since we're still breaking the signature scheme and extractable commitment, we require $T_{\mathsf{rMPC}_{(1-3)}} >> T_f$, $T_{\mathsf{rMPC}} >> T_{\mathsf{ecom}}$ and $T_{\mathsf{rMPC}} >> T_{\mathsf{Sign}}$.

Equation 41 holds from the the security of the first three rounds of the robust semi-honest MPC. Assume equations 38 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $W^{7}_{\mathrm{k}\to\mathrm{i}}$ and $W^{8}_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the security of the first three rounds of the robust semi-honest MPC. To reduce to the security of the first three rounds, as explained for 42, we expose the robust MPC an external challenger. We then break the hiding property of the corresponding two round non-malleable commitment to extract the required value. We then use D to break the security of the first three rounds of the robust semi-honest MPC. We additionally require $T_{\mathsf{rMPC}_{(1-3)}} >> T^{\mathsf{h}}_{\mathsf{nmcom}}$.

Equation 40 follows from the security of the first three rounds of the robust semi honest MPC and the fact that $T_{\mathsf{rMPC}_{(1-3)}} >> T_f$, $T_{\mathsf{rMPC}} >> T_{\mathsf{ecom}}$ and $T_{\mathsf{rMPC}} >> T_{\mathsf{Sign}}$. The latter conditions are required as we're still breaking f, the signature scheme and the extractable commitment. As before, the hybrid takes time $O(max\{T_f, T_{\mathsf{ecom}}, T_{\mathsf{Sign}}\})$ which is in turn less than T_{WIPoK}. The experiment is set up exactly as for equation 42, but instead of rewinding to extract from the non-malleable commitment, we use $\mathcal{A}$ that distinguishes the views to break the security of the robust semi-honest MPC.

$\mathbf{H_9}$: Identical to $\mathbf{H_8}$ except that we stop breaking the signature scheme and the extractable commitment, and start rewinding again to obtain the trapdoor and the adversary's inputs. We follow the same strategy on the look ahead threads as the main thread.

We claim the following

$$v^7 \approx_c v^8. \tag{43}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{7}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{8}_{\mathrm{k}\to\mathrm{i}}. \tag{44}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^{7}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{8}_{\mathrm{k}\to\mathrm{i}}. \tag{45}$$

Equation 43 follows from the fact that this was only a statistical change and the main thread remains unchanged. For the same reason, equations 44 and 45 and hold.

$\mathbf{H_{10}}$: Identical to $\mathbf{H_9}$ except that in the third round for every honest party P_{i} we send $\{r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} := (x'_{\mathrm{i}}, r'_{\mathrm{i}}) \oplus r^0_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{P_{\mathrm{i}}\in\mathcal{H}}$. Where $r^0_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}$ was the corresponding message sent inside the the extractable commitment in the first round, and $\{x'_{\mathrm{i}}, r'_{\mathrm{i}}\}_{P_{\mathrm{i}}\in\mathcal{H}}$ are the input and randomness generated and used in the first three rounds of the MPC. Thus, alternatively we view the corresponding extractable commitment scheme to contain $r^1_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} \oplus (x'_{\mathrm{i}}, r'_{\mathrm{i}})$ by making changes only in the third round (only on the main thread). This is done in two steps, which we separate as two sub-hybrids.

$\mathbf{H_{10,1=0}}$ First, we start "cheating" in the look ahead threads with respect to the extractable commitment. This is done by sending random messages (of the appropriate length) as the mask. Note that the main thread remains unchanged, and we still respond honestly to queries on the main thread.

We need the extracted values from the look ahead threads to proceed in the main thread. Thus the adversary's view on the main threads should remain unchanged. This is ensured by the rewinding security and hiding of the extractable commitment.

$$v^{9} \approx_c v^{10,0}. \tag{46}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{9}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{9,1}_{\mathrm{k}\to\mathrm{i}}. \tag{47}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^{9}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{10,0}_{\mathrm{k}\to\mathrm{i}}. \tag{48}$$

Equation 46 follows from the fact that this was only a statistical change as the main thread remains unchanged, and we are able to proceed on the main thread because we are still able to extract on the look ahead thread. For the same reasons, equations 47 and 48 and hold.

$\mathbf{H_{10}}$ Next, we change the masks in the main thread from the honest party P_{i} to be $\{r^{1}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}} := (x'_{\mathrm{i}}, r'_{\mathrm{i}}) \oplus r^{0}_{\mathsf{ecom}_{\mathrm{i}\to\mathrm{k}}}\}_{P_{\mathrm{i}}\in\mathcal{H}}$.
We claim the following

$$v^{10,0} \approx_c v^{10}. \tag{49}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad W^{10,0}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{10}_{\mathrm{k}\to\mathrm{i}}. \tag{50}$$

$$\forall P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}},\ \forall P_{\mathrm{i}} \in \mathcal{P} \qquad \widehat{W}^{10,0}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{10}_{\mathrm{k}\to\mathrm{i}}. \tag{51}$$

Assume equations 51 does not hold. Then $\exists P_{\mathrm{k}} \in \mathcal{P}^{\mathcal{A}}, P_{\mathrm{i}} \in \mathcal{H}$ such that $\widehat{W}^{10,10}_{\mathrm{k}\to\mathrm{i}}$ and $\widehat{W}^{10}_{\mathrm{k}\to\mathrm{i}}$ are distinguishable by a PPT distinguisher D. We will use this distinguisher to break the hiding property of the extractable commitment. Specifically, we rewind to extract the corresponding non-malleable commitment $\widehat{W}_{\mathrm{k}\to\mathrm{i}}$. We move from the previous hybrid to the current one by a sequence of intermediate hybrids where in each hybrid we change the value for only a single commitment (by its mask). If equation 51 does not hold, then two adjacent hybrids are distinguishable. Let equation 51 not hold when we change the masked value in $\pi_{\mathsf{ecom}_{\mathrm{i}^*\to\mathrm{k}^*}}$. To reduce to the hiding property of the extractable commitment, we expose $\pi_{\mathsf{ecom}_{\mathrm{i}^*\to\mathrm{k}^*}}$ to an external challenger. All messages other than those of $\pi_{\mathsf{ecom}_{\mathrm{i}^*\to\mathrm{k}^*}}$ are computed as in the previous hybrid. As challenge we send $(x_{\mathrm{i}}, r_{\mathrm{i}}) \oplus r$ and $(x'_{\mathrm{i}}, r'_{\mathrm{i}}) \oplus r$ where r is a random value. And the mask sent in the third round is r. Depending on the value committed by the external challenger we are in one of the two adjacent hybrids. We then rewind the third and fourth rounds of the non-malleable commitments inside the NMZK to extract the masked value. Since only the third round of the commitment scheme overlaps with the rewinding, we send the same message in each look-ahead thread. We then use D to break the hiding property of the extractable commitment. We require $T_{\mathsf{ecom}} >> T_f$ as we're breaking the OWP. Following from the previous sub-hybrid, we are still "cheating" on the look ahead threads when we rewind to extract.

Equation 50 holds trivially from the fact that changes are made in the third round after completion of the two round non-malleable commitment.

Equation 49 follows from the hiding property of the extractable commitment, and the fact that $T_{\mathsf{ecom}} >> T_f$. The latter is required as we're still breaking f to simulate the input delayed WIPoK. The experiment is set up exactly as for equation 51, but instead of rewinding to extract from the non-malleable commitment, we use $\mathcal{A}$ that distinguishes the views to break the witness indistinguishability.

$\mathbf{H_{11}}$**:** Identical to $\mathbf{H_{10}}$ except that we stop rewinding to extract the input, and instead break the signature scheme and the extractable commitment to obtain the trapdoor and the adversary's inputs.

We claim the following

$$v^{10} \approx_c v^{11}. \tag{52}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^{10}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{11}_{\mathrm{k}\to\mathrm{i}}. \tag{53}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad \widehat{W}^{10}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{11}_{\mathrm{k}\to\mathrm{i}}. \tag{54}$$

Equation 52 follows from the fact that this was only a statistical change as the main thread remains unchanged. For the same reason, equations 53 and 54 and hold.

$\mathbf{H_{12}}$**:** Identical to $\mathbf{H_{11}}$ except that we use the valid witness to complete the input delayed WIPoK proofs.

We claim the following

$$v^{11} \approx_c v^{12}. \tag{55}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^{11}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{12}_{\mathrm{k}\to\mathrm{i}}. \tag{56}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad \widehat{W}^{11}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{12}_{\mathrm{k}\to\mathrm{i}}. \tag{57}$$

The claims follow identically from hybrid $\mathbf{H_7}$.

$\mathbf{H_{13}}$**:** Identical to $\mathbf{H_{12}}$ except that we stop breaking the one-way permutation to obtain the trapdoor and commit to 0 in the two round non-malleable commitment.

We claim the following

$$v^{12} \approx_c v^{13}. \tag{58}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^{12}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{13}_{\mathrm{k}\to\mathrm{i}}. \tag{59}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad \widehat{W}^{12}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{13}_{\mathrm{k}\to\mathrm{i}}. \tag{60}$$

The claims follow identically from hybrid $\mathbf{H_5}$.

$\mathbf{H_{14}}$**:** Identical to $\mathbf{H_{13}}$ except that we stop breaking the signature scheme and the extractable commitment, and start rewinding again to obtain the trapdoor and the adversary's inputs. In the look ahead threads, we follow the same strategy as the main thread (i.e. we no longer "cheat" on the look ahead threads). Note that in this hybrid we're running in **polynomial time** again.

We claim the following

$$v^{13} \approx_c v^{14}. \tag{61}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad W^{13}_{\mathrm{k}\to\mathrm{i}} \approx_c W^{14}_{\mathrm{k}\to\mathrm{i}}. \tag{62}$$
$$\forall P_\mathrm{k} \in \mathcal{P}^\mathcal{A},\ \forall P_\mathrm{i} \in \mathcal{P} \qquad \widehat{W}^{13}_{\mathrm{k}\to\mathrm{i}} \approx_c \widehat{W}^{14}_{\mathrm{k}\to\mathrm{i}}. \tag{63}$$

Equation 61 follows from the fact that this was only a statistical change and the main thread remains unchanged. For the same reason, equations 62 and 63 and hold.

Hybrid $\mathbf{H_{14}}$ is identical to our simulator. From the above discussion, we have

$$v^0 \approx_c v^{14}$$

thus proving security of the constructed MPC.

This completes the security proof.