Four-Round Secure Multiparty Computation from General Assumptions

Michele Ciampi
The University of Edinburgh
UK
mciampi@ed.ac.uk

Rafail Ostrovsky
UCLA
Los Angeles
rafail@cs.ucla.edu

Abstract

In this work we continue the study on the round complexity of secure multi-party computation with black-box simulation in the simultaneous broadcast model where all the parties get the output.

In Eurocrypt 2016 Garg at al. show that four rounds are necessary to obtain a secure multi-party computation protocol for any function in the plain model. Many different works have tried to show that, relying on standard assumptions, four rounds are also sufficient for MPC. In Crypto 2017 Ananth et al. and in TCC 2017 Brakerski at al. propose a four-round protocol based on quasi-polynomial time number theoretic assumptions. In Crypto 2018 the two independent works of Badrinarayanan et al. and Halevi at al. show how reach the four-round barrier relying on number theoretic polynomial-time assumptions.

In this work we propose a compiler that takes as input a three-round oblivious transfer protocol, and outputs a four-round MPC protocol. Our compiler is also based on two-round witness indistinguishable proof (zap). We also show how to obtain three-round OT assuming sub-exponentially secure trapdoor permutations and zap. As a corollary we obtain the first four-round MPC protocol that relies on general assumptions. Moreover, given the three-round OT from claw-free trapdoor permutation proposed by Hazay at el. in SCN 2016, we obtain the first four-round MPC protocol that is based on general polynomial-time assumptions.

1 Introduction

Obtaining round-optimal secure computation [Yao82, GMW87] has been a long standing open problem. In [KOS03], Katz et al. obtained a constant-round secure multi-party computation (MPC) protocol using sub-exponential hardness assumptions. This result was then improved by Pass in [Pas04] that showed how to get bounded-concurrent secure MPC for any functionality with standard assumptions. Further results of Goyal [Goy11] and Goyal et al. [GLOV12] relied on better assumptions but with a round complexity still far from optimal.

In Eurocrypt '16 Garg et al. [GMPP16] makes an important jump ahead towards fully understanding the round complexity of secure MPC computation showing the 4 rounds are necessary to compute any two-party functionality where both the parties get the output. In Crypto '17 Ananth et al. [ACJ17] and a concurrent and independent work in TCC '17 of Brakerski at al. [BHP17], propose the first 4-round MPC protocol for any functionality assuming number theoretic assumption w.r.t. superpolynomial-time adversaries. In Eurocrypt '18, Benhamouda et al. [BL18] show how to promote any k-round OT protocol to a k-round MPC secure protocol using interactive garbled

	#Rounds	Assumptions
[ACJ17]	5	DDH
[ACJ17]	4	OWP+Sub-exp DDH
[BHP17]	4	Sub-exp LWE
[BL18]	5	5-round OT
[BGJ$^+$18]	4	DDH or QR or N$^{\text{th}}$-Residuosity
[HHPV18]	4	LWE or DDH or QR or DCR
This work	4	3-round OT + zap + OWF

Table 1: Comparison with existent works

circuit (a similar technique is proposed in [GS18]).[1] The very recent works of Badrinarayanan et al. [BGJ$^+$18] and Halevi et. al. [HHPV18] presented at Crypto '18 propose a 4-round protocol that is secure under the DDH assumption (and other number theoretic assumptions, see Tab. 1 for more details). However, all the works leave open the following question:

Open Question: is there a 4-round secure MPC protocol under general assumptions?

1.1 Our Contributions

This paper answers to the above question introducing a compiler that takes as input a 3-round oblivious transfer (OT) protocol and outputs a 4-round MPC secure protocol. Our compiler relies on the only additional assumption that zap and one-way functions (OWFs) exist, which in turns implies that our 4-round MPC protocol can be constructed from 3-round OT and certified trapdoor permutations (TDPs) (see [DN00] for more details on the minimal assumptions required to obtain zap). The 3-round OT required for our construction needs to enjoy only a relaxed form of security called *delayed-semi-malicious security*. That is, the security of the protocol holds only if the adversary can output a *defense* in the second round that proves his honest behaviour. A defense is represented by a randomness and an input that proves that the messages computed by the adversary are consistent with the protocol description. As an additional contribution of this work, we show how to obtain 3-round OT that is secure against malicious adversaries (and also against delayed-semi-malicious adversary) assuming that trapdoor permutations and zap exist. More precisely, we prove that our OT protocol is *private*. That is, a malicious receiver cannot figure out if the sender has used the input (l_0, l_1) or (l_b, l_b) where b corresponds to the input bit of the receiver. Similarly, a malicious sender should not be able to distinguish whether the receiver uses the bit 0 or the bit 1. By combining this result with our MPC compiler we obtain the first 4-round MPC protocol from sub-exponentially secure trapdoor permutations[2]. Moreover, the work of Hazay et al. [HV16] proposes a 3-round private OT protocol assuming that claw-free trapdoor permutations exist. This yields a 4-round maliciously secure MPC protocol that is provable secure under the assumption that claw-free trapdoor permutations and zap exist thus obtaining the first 4-round MPC protocol that is secure under general polynomial-time hardness assumptions. In summary, in this work we prove three main theorems:

[1] The OT required by the construction proposed in [BL18] has to enjoy only a mild form of security instead of the standard simulation based security. More details follow.

[2] We note that zap can be instantiated from certified trapdoor permutations.

Theorem 1 (informal). *If sub-exponentially secure trapdoor permutations and zap exist then there is a 3-round private OT protocol.*

Theorem 2 (informal). *If sub-exponentially secure trapdoor permutations and zap exist then there is a 4-round MPC protocol.*

Theorem 3 (informal). *If claw-free trapdoor permutations and zap exist then there is a 4-round MPC protocol.*

1.2 Our Techniques

In this section we provide intuitions and high level description of our techniques.

1.2.1 Round-optimal MPC.

Our starting point is the main result of [BL18] that presents a compiler that transforms a k-round semi-malicious OT protocol into a k-round delayed-semi-malicious (DSM) MPC protocol. A delayed-semi-malicious MPC protocol is a secure MPC protocol in which the security holds only if the adversary outputs in the second last round a valid defense that explains the messages he has computed. In [BL18] it is also showed how to obtain a 5-round secure MPC protocol starting from a 4-round delayed-semi-malicious one. The 5-round MPC protocol proposed by Benhamouda at al. works as follows. Each party runs a 4-round delayed-input non-malleable zero-knowledge argument NMZK in parallel with the first three rounds of a DSM MPC protocol $\mathsf{MPC}^{\mathsf{DSM}}$ thus proving that the first three rounds of $\mathsf{MPC}^{\mathsf{DSM}}$ are correctly computed ($\mathsf{MPC}^{\mathsf{DSM}}$ starts in the second round). Moreover, an additional execution of NMZK is run from the second to the fifth round to prove that also the fourth round of $\mathsf{MPC}^{\mathsf{DSM}}$ is well formed. This approach yields to a 5-round protocol (each execution of NMZK requires 4-rounds, and the two execution are shifted by one round). In order to reduce the round complexity of the protocol, we get rid of the first execution of NMZK, and prove the correctness of the second round of $\mathsf{MPC}^{\mathsf{DSM}}$ using a two-round resettable witness-indistinguishable proof referred as zap in [DN00]. Moreover, we require the parties to commit to a valid defense using a three-round non-malleable commitment NMCOM[3]. In summary, our protocol works as follows. In the first three rounds each party: 1) commits to a valid defense using NMCOM, 2) commits to a random values alway using NMCOM, 3) runs the first two rounds of $\mathsf{MPC}^{\mathsf{DSM}}$ and 4) proves using zap that either the first non-malleable commitment contains a valid defense for the messages of $\mathsf{MPC}^{\mathsf{DSM}}$, or that a trapdoor is committed in the second non-malleable commitment. In these first three rounds each party also starts executing a NMZK that is used to prove that the last message of $\mathsf{MPC}^{\mathsf{DSM}}$ (which is sent in the fourth round) is well formed. Always in the first three rounds a special witness-indistinguishable proof of knowledge (WIPoK) WIPoK is run in which each party proves the knowledge of a secret information (which will be used as a trapdoor by the simulator). In the last round only the fourth round of NMZK and the third round of $\mathsf{MPC}^{\mathsf{DSM}}$ are sent (see Fig. 3 for the high level description of the protocol).

In the security proof we can show that even though the simulator is committing to a trapdoor (which is extracted by rewinding WIPoK) using the second flow of NMCOM, the adversary cannot do the same due to the non-malleability of NMCOM and NMZK. From the soundness of zap this implies that the adversary is committing to a valid defense in the first flow of NMCOM, which means that the

[3] For our construction we need to use a non-malleable commitment that is only one-one. Even though this seems to be counterintuitive since we are considering a multi-party computation protocol, we show that one-one non-malleability is enough for our purpose.

simulator can extract the defense from the non-malleable commitment by rewinding NMCOM from the third to the second message. We note that we require the non-malleable commitment scheme to be only *honest-extractable*. That is, the extractor outputs the actual committed value only if the committer is behaving honestly. We observe that the adversary is forced to behave correctly due to the soundness of the zap. However, the security proof we just sketched has a subtlety. The simulator needs to obtain the trapdoor to be committed in the second flow of NMCOM during the first three rounds of the protocol. Since the first chance that the simulator has to extract such a trapdoor is by rewinding from the third to the second round of WIPoK and since he needs to complete the zap proof in the third round using a valid witness, then the simulator needs another witness to run the zap. Indeed, the first time that the simulator interacts with the adversary he does not know the trapdoor and therefore he cannot commit to it using the non-malleable commitment. To circumvent this limitation we let the simulator interact with the adversary using the *honest witness* for the zap (the defence for $\mathsf{MPC}^{\mathsf{DSM}}$). That is, the simulator takes a random input x' and a randomness ρ to run the first two rounds of $\mathsf{MPC}^{\mathsf{DSM}}$. Then, he commits to (x', ρ) using NMCOM, and uses the knowledge of (x', ρ) and of the decommitment information of NMCOM to execute the zap. This strategy is used in the *look-ahead threads* in order to extract the trapdoor from WIPoK. Once the trapdoor is extracted, the simulator goes back to the second round, and: 1) simulates the messages of $\mathsf{MPC}^{\mathsf{DSM}}$, 2) commits to the trapdoor using NMCOM and 3) runs the zap using the knowledge of the trapdoor. After that, the simulator sends the last round of NMZK and $\mathsf{MPC}^{\mathsf{DSM}}$ thus concluding the interaction with the adversary. A similar technique is used in [HHPV18] where the rewinds made to extract the trapdoor are called *premature rewinds*. Given how delicate such part of the proof is, we propose a generalisation of this proof technique which can be seen as an additional contribution of this work (see Sec. 4 for more details).

1.2.2 Three-round oblivious transfer.

We start by considering the classical semi-honest 3-round OT protocol Π^{sh} from trapdoor permutations. Let (l_0, l_1) be the input of the sender and b be the input of the receiver. In the first round of Π^{sh} the sender samples two trapdoor permutations (f_0, f_1) (with $f_d : \{0,1\}^\lambda \rightarrow \{0,1\}^\lambda$ with for $d = 0, 1$) with the corresponding trapdoors and sends (f_0, f_1) to the receiver. The receiver takes a random value x and computes $f_b(x) \leftarrow X_b$. Then she takes another random value X_{1-b} and sends (X_0, X_1) to the sender. Using the trapdoors, the sender computes $W_d = f^{-1}(X_d) \oplus l_d$ for $d = 0, 1$ and sends (W_0, W_1) to the receiver. The receiver then computes $l_b = x \oplus W_b$. Clearly, in the case that the receiver is malicious there is nothing that prevents him to get both l_0 and l_1. We modify Π^{sh} in order to be resilient against a malicious receiver. In our protocol, in addition to (f_0, f_1), the sender sends also a random string $A \in \{0,1\}^\lambda$ and the first round of zap. The receiver then takes a random value x and computes $f_b(x) \leftarrow Y_b$ (as in Π^{sh}). Then she picks $k \leftarrow \{0,1\}^{\lambda_k}$ with $\lambda_k = \lambda/c$ for a constant $c \geq 2$, and computes $X_{1-b} = \mathsf{PRG}(k) \oplus A$, where PRG is a one-to-one pseudo-random generator $\mathsf{PRG} : \{0,1\}^{\lambda_k} \rightarrow \{0,1\}^\lambda$. The receiver then sends (X_0, X_1) and completes the zap proving that either $X_0 = \mathsf{PRG}(k) \oplus A$ or $X_1 = \mathsf{PRG}(k) \oplus A$. The sender checks that the zap verifies, computes $W_d = f^{-1}(X_d) \oplus l_d$ for $d = 0, 1$ and sends (W_0, W_1) to the receiver (as in Π^{sh}). The security against a malicious sender comes almost immediately from the security of the PRG and the witness-indistinguishability of zap. Intuitively, the security against a malicious receiver comes from the fact that the adversary, in order to obtain also l_{1-b}, should find a value k such that $\mathsf{PRG}(k) \oplus A$ corresponds to a value X_{1-b} that he knows how to invert. Note that we are giving the adversary the power to decide (to some extent) what X_{1-b} should be. However, since the length of k is shorter (by a constant factor) than the length of A, the adversary can choose X_{1-b} from a set that is exponentially smaller than the domain of the trapdoor permutation. For the proof to work we need to assume the trapdoor permutation to be sub-exponentially secure. That is, we require that no probabilistic polynomial-time adversary can invert trapdoor permutation with probability larger than $2^{-\lambda}$.[4]

2 Definitions and Tools

Preliminaries. We denote the security parameter by λ and use "$||$" as concatenation operator (i.e., if a and b are two strings then by $a||b$ we denote the concatenation of a and b). For a finite set Q, $x \leftarrow Q$ denotes a sampling of x from Q with uniform distribution. We use the abbreviation PPT that stands for probabilistic polynomial time. We use $\mathsf{poly}(\cdot)$ to indicate a generic polynomial function.

Let A and B be two interactive probabilistic algorithms. We denote by $\langle A(\alpha), B(\beta)\rangle(\gamma)$ the distribution of B's output after running on private input β with A using private input α, both running on common input γ. Typically, one of the two algorithms receives 1^λ as input. A *transcript* of $\langle A(\alpha), B(\beta)\rangle(\gamma)$ consists of the messages exchanged during an execution where A receives a private input α, B receives a private input β and both A and B receive a common input γ. Moreover, we will refer to the *view* of A (resp. B) as the messages it received during the execution of $\langle A(\alpha), B(\beta)\rangle(\gamma)$, along with its randomness and its input. We say that the transcript τ of an execution $b = \langle \mathcal{P}(z), \mathcal{V}\rangle(x)$ is *accepting* if $b = 1$. We say that a protocol (A, B) is public coin if B sends to A random bits only. When it is necessary to refer to the randomness r used by and algorithm A we use the following notation: $A(\cdot; r)$.

2.1 Standard Definitions

Definition 1 (Trapdoor permutation). *Let $\mathcal{F}$ be a triple of* PPT *algorithms* $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ *such that if* $\mathsf{Gen}(1^\lambda)$ *outputs a pair* $(f, \mathtt{td})$*, then* $\mathsf{Eval}(f, \cdot)$ *is a permutation over* $\{0,1\}^\lambda$ *and Invert* $(f, \mathtt{td}, \cdot)$ *is its inverse.* $\mathcal{F}$ *is a trapdoor permutation such that for all* PPT *adversaries* $\mathcal{A}$*:*

$$\mathrm{Prob}\left[\ (f, \mathtt{td}) \leftarrow \mathsf{Gen}(1^\lambda); y \leftarrow \{0,1\}^\lambda, x \leftarrow \mathcal{A}(f, y) : \mathsf{Eval}(f, x) = y\ \right] \leq \nu(\lambda).$$

For convenience, we drop $(f, \mathtt{td})$ from the notation, and write $f(\cdot)$, $f^{-1}(\cdot)$ to denote algorithms $\mathsf{Eval}(f, \cdot)$, $\mathsf{Invert}(f, \mathtt{td}, \cdot)$ respectively, when f, $\mathtt{td}$ are clear from the context. Following [KO04, GMPP16] we assume that $\mathcal{F}$ satisfies (a weak variant of) "certifiability": namely, given some f it is possible to decide in polynomial time whether $\mathsf{Eval}(f, \cdot)$ is a permutation over $\{0,1\}^\lambda$. Let hc be the hardcore bit function for λ bits for the family $\mathcal{F}$. λ hardcore bits are obtained from a single-bit hardcore function h and $f \in \mathcal{F}$ as follows: $\mathsf{hc}(z) = h(z)||h(f(z))||\ldots||h(f^{\lambda-1}(z))$. Informally, $\mathsf{hc}(z)$ looks pseudorandom given $f^\lambda(z)$[5].

[4] In our scheme we actually use the hardcore bit function of f and rely on the assumptions that no polynomial-time adversary can distinguish a random string from the output of the hardcore bit function with probability greater than $\frac{1}{2} + 2^{-\lambda}$.

[5] $f^\lambda(z)$ means the λ-th iteration of applying f on z.

2.2 (Delayed-Input) Proof/Argument Systems

Definition 2 (Proof/argument system)**.** *A pair of* PPT *interactive algorithms* $\Pi = (\mathcal{P}, \mathcal{V})$ *constitute a* proof system *(resp., an* argument system*) for an* $\mathcal{NP}$*-language* L*, if the following conditions hold:*

Completeness: *For every* $x \in L$ *and* w *such that* $(x, w) \in \mathsf{Rel_L}$*, it holds that:*

$$\text{Prob}\,[\,\langle\mathcal{P}(w), \mathcal{V}\rangle(x) = 1\,] = 1.$$

Soundness*: For every interactive (resp.,* PPT *interactive) algorithm* $\mathcal{P}^\star$*, there exists a negligible function* ν *such that for every* $x \notin L$ *and every* z*:*

$$\text{Prob}\,[\,\langle\mathcal{P}^\star(z), \mathcal{V}\rangle(x) = 1\,] < \nu(|x|).$$

A proof/argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for an $\mathcal{NP}$-language L, enjoys *delayed-input* completeness if $\mathcal{P}$ needs x and w only to compute the last round and $\mathcal{V}$ needs x only to compute the output. Before that, $\mathcal{P}$ and $\mathcal{V}$ run having as input only the size of x. The notion of delayed-input completeness was defined in [CPS+16]. For a protocol that enjoys delayed-input completeness we consider also the notion of adaptive-input arguments/proof system. That is, the soundness holds against a stronger adversary $\mathcal{P}^*$ that can choose the statement to be proved in the last round of the interaction with $\mathcal{V}$. Analogously we also consider the notion of adaptive-input arguments/proof of knowledge (see App. A for more details).

Definition 3 (Special Honest Verifier Zero-knowledge (Special HVZK))**.** *A 3-round proof system* $\Pi = (\mathcal{P}, \mathcal{V})$ *is Special HVZK if there exists a* PPT *simulator algorithm* Sim *that for any* $x \in L$*, security parameter* λ *and second round* c *works as follow:* $(\mathsf{a}, \mathsf{z}) \leftarrow \mathsf{Sim}(1^\lambda, x, \mathsf{c})$*. Furthermore, the distribution of the output of* Sim *is computationally indistinguishable from the distribution of a transcript obtained when* $\mathcal{V}$ *sends* c *as challenge and* $\mathcal{P}$ *runs on common input* x *and any* w *such that* $(x, w) \in \mathsf{Rel_L}$*.*

In our MPC construction we use the three-round public-coin Special HVZK proof system proposed in [Blu86] which can be instantiated from one-to-one OWFs.

Definition 4 (Witness Indistinguishable (WI))**.** *An argument/proof system* $\Pi = (\mathcal{P}, \mathcal{V})$*, is* Witness Indistinguishable (WI) *for a relation* Rel *if, for every malicious* PPT *verifier* $\mathcal{V}^\star$*, there exists a negligible function* ν *such that for all* x, w, w' *such that* $(x, w) \in \mathsf{Rel}$ *and* $(x, w') \in \mathsf{Rel}$ *it holds that:*

$$\left|\text{Prob}\,[\,\langle\mathcal{P}(w), \mathcal{V}^\star\rangle(x) = 1\,] - \text{Prob}\left[\,\langle\mathcal{P}(w'), \mathcal{V}^\star\rangle(x) = 1\,\right]\right| < \nu(|x|).$$

Obviously one can generalize the above definitions of WI to their natural adaptive-input variants, where the adversarial verifier can select the statement and the witnesses adaptively, before the prover plays the last round.

As a WI argument system we make use of ZAP. ZAPs are two-round witness indistinguishable delayed-input proof systems [DN00]. As noted in [DN00] by using pseudo-random functions, ZAPs are also resettably-witness-indistinguishable. In [DN00] it is showed how to obtain ZAP from multiple certified trapdoor permutations.

2.3 Non-Malleable Commitment Schemes

A commitment scheme involves two players: sender and receiver. Informally, it consists of two phases, a commitment phase and a decommitment phase. In the commitment phase the sender, with a secret input m, interacts with the receiver. In the end of this interaction we say that a *commitment* of the message m has been computed. Moreover the receiver still does not know what m is (i.e. m is hidden) and at the same time the sender can subsequently (i.e., during the decommitment phase) open this commitment only to m (see Def. 19 in App. A for a formal definition of commitment scheme).

In order to define a non-malleable commitment we follow [LPV08, LPV09]. Let $\Pi = (\mathsf{Sen}, \mathsf{Rec})$ be a statistically binding commitment scheme. And let λ be the security parameter. Consider a MiM adversary $\mathcal{A}$ that, on auxiliary input z participates in a left and a right session. In the left sessions the MiM adversary $\mathcal{A}$ interacts with Sen receiving commitment to value m using an identity $\mathtt{id}$ of its choice. In the right session $\mathcal{A}$ interacts with Rec attempting to commit to a related value $\tilde{m}$ again using identity of its choice $\tilde{\mathtt{id}}$. If the right commitment is invalid, or undefined, its value is set to $\perp$. Furthermore, if $\tilde{\mathtt{id}} = \mathtt{id}$ then $\tilde{m}$ is also set to $\perp$ (i.e., a commitment where the adversary uses the same identity of the honest senders is considered invalid). Let $\mathsf{mim}_{\Pi}^{\mathcal{A},m}(z)$ denote a random variable that describes the values $\tilde{m}$ and the view of $\mathcal{A}$ in the above experiment.

Definition 5 (One-one non-malleable commitment scheme [LPV08, LPV09])**.** *A commitment scheme is* non-malleable with respect to commitment *if, for every* PPT *MiM adversary $\mathcal{A}$, for every $m_0 \in \{0,1\}^{\mathsf{poly}(\lambda)}$ and $m_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following holds*

$$\{\mathsf{mim}_{\Pi}^{\mathcal{A},m_0}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{mim}_{\Pi}^{\mathcal{A},m_1}(z)\}_{z\in\{0,1\}^\star}.$$

We say that a commitment is valid or well formed if it admits a decommitment to a message $m \neq \perp$. When the identity is selected by the sender then the above id-based definitions guarantee non-malleability without ids as long as the MiM does not behave like a proxy (an unavoidable attack). Indeed the sender can pick as $\mathtt{id}$ the public key of a strong signature scheme signing the transcript. The MiM will have to use a different $\mathtt{id}$ or to break the signature scheme. To not overburden the notation we sometimes omit the ids in the formal description of our protocols. In this work we also consider the notion of many-one NM commitment scheme in which $\mathcal{A}$ participates in one right session and polynomially-many left sessions. Following [LP11] we say that a MiM is *synchronous* if it "aligns" the left and the right sessions; that is, whenever it receives message i on the left, it directly sends message i on the right, and vice versa.

Definition 6 (Synchronous NM commitments)**.** *A NM commitment scheme* Π *is synchronous if its security holds only against synchronous MiM.*

Using the approach proposed in the security proof of Proposition 1 provided in [LPV08], it is possible to claim that a synchronous (one-one) non-malleable commitment is also synchronous many-one non-malleable.

2.3.1 3-Round Honest-Extractable Commitment Schemes.

Informally, a 3-round commitment scheme is honest-extractable if there exists an efficient extractor that having black-box access to any efficient honest sender that successfully performs the commitment phase, outputs the only committed string that can be successfully decommitted. We give now a definition that follows the one of [PW09].

Definition 7 (Honest-Extractable Commitment Scheme)**.** *A perfectly (resp. statistically) binding commitment scheme* $\mathtt{ExCS} = (\mathrm{ExSen}, \mathrm{ExRec})$ *is an* honest-extractable commitment scheme *if there exists an expected* PPT *extractor* $\mathtt{ExtCom}$ *that given oracle access to any honest sender* ExSen*, outputs a pair* (τ, m) *such that the following two properties hold:*

- **Simulatability:** *τ is identically distributed to the view of* ExSen *(when interacting with an honest* ExRec*) in the commitment phase.*
- **Extractability:** *the probability that there exists a decommitment of τ to a message m', where $m' \neq m$ is 0 (resp. negligible).*

2.3.2 Special non-malleable commitment scheme.

For our propose we use a 3-round *special non-malleable commitment scheme*. That is, a commitment scheme that 1) is one-one synchronous non-malleable, 2) is honest-extractable, 3) has the property of last-message pseudorandomness, 4) is delayed-input and 5) has reusable decommitment information. We now give more details of these properties and on how they can be achieved.

The property of *last-message pseudorandomness* is defined in [BGJ+18, Definition 4]. Informally this property states that a malicious receiver cannot distinguish between a well formed commitment of a message m, and a commitment where only the first two rounds are well formed and the third is a random string. The authors of [BGJ+18] states that the property of last-message pseudorandomness is enjoyed by the scheme proposed in [GPR16]. The non-malleable commitment Π provided in Figure 2 of [GPR16] enjoys non-malleability against synchronous adversary (as proved in Theorem 1 of [GPR16]), and can be instantiated in three rounds using one-to-one OWFs). Also, as stated in Section 5 of [GPR16], given a commitment computed by the sender of Π one can rewind the sender in order to obtain a new accepting transcript with the same first round (resp., first two rounds if we consider the instantiation that relies on OWFs) in order to extract a decommitment information d. Moreover, if the sender is honest, then it is possible to use d to extract the actual message committed by the sender (we remark that we do not require any form of extractability against malicious senders). A delayed-input commitment scheme is a commitment scheme that retains its security (binding, hiding and non-malleability) even if the adversary can decide one of the challenge messages in the last round. In [COSV16] it is showed that any commitment scheme can be made delayed-input. The property of *reusable decommitment information* ensures that, when the sender is honest, the decommitment information d for a commitment τ extracted by rewinding the sender (ad discussed above) can be re-used to extract the message committed in τ' where τ and τ' share the same first round. It is easy to see that this property is enjoyed by the construction proposed in [GPR16]. We refer the reader to App. B for the description of the scheme proposed in [GPR16] and more details on the properties on honest-extractability and reusable decommitment information.

Corollary 1. *The synchronous 3-round non-malleable commitment scheme proposed in [GPR16] is a* special non-malleable commitment scheme*.*

2.4 Delayed-Input Non-Malleable Zero Knowledge

Here we follow [COSV17]. The definition of [COSV17] allows the adversary to explicitly select the statement, and as such the adversary provides also the witness for the prover. The simulated game however will filter out the witness so that the simulator will receive only the instance. This approach strictly follows the one of [SCO+01] where adaptive-input selection is explicitly allowed and managed in a similar way. As final remark, this definition will require the existence of a black-box simulator since a non-black-box simulator could retrieve from the code of the adversary the witness for the adaptively generated statement. The non-black-box simulator could then run the honest prover procedure, therefore canceling completely the security flavor of the simulation paradigm.

Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a delayed-input interactive argument system for a $\mathcal{NP}$-language L with witness relation $\mathsf{Rel_L}$. Consider a PPT MiM adversary $\mathcal{A}$ that is simultaneously participating in one left session and $\mathsf{poly}(\lambda)$ right sessions. Before the execution starts, $\mathcal{P}, \mathcal{V}$ and $\mathcal{A}$ receive as a common input the security parameter in unary 1^λ. Additionally $\mathcal{A}$ receives as auxiliary input $z \in \{0,1\}^\star$. In the left session $\mathcal{A}$ verifies the validity of the prove given by $\mathcal{P}$ with respect to the statement x (chosen adaptively in the last round of Π). In the right sessions $\mathcal{A}$ proves the validity of the statements $\tilde{x}_1, \dots, \tilde{x}_{\mathsf{poly}(\lambda)}$[6] (chosen adaptively in the last round of Π) to the honest verifiers $\mathcal{V}_1, \dots, \mathcal{V}_{\mathsf{poly}(\lambda)}$.

More precisely in the left session $\mathcal{A}$, before the last round of Π is executed, adaptively selects the statement x to be proved and the witness w, s.t. $(x, w) \in \mathsf{Rel_L}$, and sends them to $\mathcal{P}$.

Let $\mathsf{View}^{\mathcal{A}}(1^\lambda, z)$ denote a random variable that describes the view of $\mathcal{A}$ in the above experiment.

Definition 8 (Delayed-input NMZK)**.** *A delayed-input argument system* $\Pi = (\mathcal{P}, \mathcal{V})$ *for an* $\mathcal{NP}$*-language* L *with witness relation* $\mathsf{Rel_L}$ *is delayed-input non-malleable zero knowledge (NMZK) if for any MiM adversary* $\mathcal{A}$ *that participates in one left session and* $\mathsf{poly}(\lambda)$ *right sessions, there exists a expected* PPT *machine* $S(1^\lambda, z)$ *such that:*

1. *Let* $(\mathsf{View}, w_1, \dots, w_{\mathsf{poly}(\lambda)})$ *denote the output of* $S(1^\lambda, z)$*, for some* $z \in \{0,1\}^\star$*. The probability ensembles* $\{S^1(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^\star}$ *and* $\{\mathsf{View}^{\mathcal{A}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^\star}$ *are computationally indistinguishable over* λ*, where* $S^1(1^\lambda, z)$ *denotes the first output of* $S(1^\lambda, z)$*.*
2. *For every* $i \in \{1, \dots, \mathsf{poly}(\lambda)\}$*, if the* i*-th right session is accepting w.r.t. some statement* x_i *and* $\mathcal{A}$ *does not acts as a proxy (by simply sending back and forward the massages of the left session), then* w_i *is s.t.* $(x_i, w_i) \in \mathsf{Rel_L}$*.*

The above definition of NMZK allows the adversary to select statements adaptively in the last round both in left and in the right sessions. Therefore any argument system that is NMZK according to the above definition enjoys also adaptive-input argument of knowledge. Such a delayed-input NMZK argument system can be instantiated from one-to-one OWFs as showed in [COSV17]. In this work we assume that the simulator-extractor works by internally rewinding the adversary from the third to the second round and from the fourth to the third round in order to extract the witnesses for the theorems proved by the adversary. The simulator for the scheme proposed in [COSV17] follows this strategy.

2.5 Oblivious Transfer

In this section we consider the definition for OT in presence of malicious sender and receiver shown in [HV16]. In the definitions presented below, we denote the honest sender and receiver algorithms by S and R respectively. Recall that the oblivious transfer functionality is specified by the function $F_{\mathcal{OT}} : ((l_0, l_1), b) \mapsto (-, l_b)$ that takes as input (l_0, l_1) from the sender, a bit b from the receiver, and outputs l_b to the receiver. We say that a protocol $\Pi = (S, R)$ realizes the OT functionality if the protocol computes $F_{\mathcal{OT}}$ correctly. As in [HV16], we focus on the case of a three-round protocol where the sender sends the first message. Furthermore, our definition will restrict the honest sender's algorithm to be described by a pair of algorithms $S = (S_1, S_2)$ where S_1 on input 1^λ outputs the first message m_1^s of the OT protocol and state σ and S_2 on input $(\sigma, m^r, (l_0, l_1))$ generates the third message of the OT protocol where m^r is the response from the receiver and (l_0, l_1) is the sender's input.

[6] We denote (here and in the rest of the paper) by $\tilde{\delta}$ a value associated with the right session where δ is the corresponding value in the left session.

Definition 9 (Sender's privacy [HV16]). *A protocol Π that realizes the $F_{\mathcal{OT}}$ is private with respect to a malicious receiver if for any PPT adversary $R^\star$ and PPT distinguisher D there exists a negligible function $\nu(\cdot)$ such that for all λ, except with probability $\nu(\lambda)$ over $((m_1^s, m^r), \sigma, r_R)$, where r_R is the randomness used by $R^\star$, there exists a bit b, such that for any strings (l_0, l_1) and auxiliary input z,*

$$\begin{aligned} |\Pr\{m_2^s \leftarrow S_2(\sigma, m_r, (l_0, l_1)) : D(1^\lambda, z, r_R, (m_1^s, m_2^s)) = 1\} - \\ \Pr\{x_b \leftarrow l_b; x_{1-b} \leftarrow \{0,1\}^\lambda; m_2^s \leftarrow S_2(\sigma, m_r, (x_0, x_1)) : \\ D(1^\lambda, z, r_R, (m_1^s, m_2^s)) = 1\}| \leq \nu(\lambda) \end{aligned}$$

Let $\langle S^\star(1^\lambda), R(b)\rangle(1^\lambda)$ denote the random variable describing the corrupted sender's output when interacting with R that is invoked on inputs b.

Definition 10 (Receiver's privacy [HV16]). *A protocol Π that realizes the $F_{\mathcal{OT}}$ is private with respect to a malicious sender if for any PPT adversary $S^\star$ corrupting S the following holds*

$$\{View_{S^\star}[\langle S^\star(1^\lambda), R(1^\lambda, 0)\rangle]\} \approx \{View_{S^\star}[\langle S^\star(1^\lambda), R(1^\lambda, 1)\rangle]\}$$

We say that an OT protocol is *private* if it is both sender and receiver private.

2.6 MPC Definitions

In this section we give the definition of malicious secure MPC and delayed-semi-malicious MPC following [BL18]. We consider the MPC protocols where at each round ℓ, each party P_i broadcasts a message $\mathtt{msg}_i^\ell$ to all the other parties simultaneously.

Definition 11 (MPC protocol [BL18]). *Let n be a positive integer, m a polynomial in the security parameter, and f an n party-functionality. An m-round MPC protocol* MPC *for f is a tuple of deterministic polynomial-time algorithms* $\mathsf{MPC} = (\mathsf{Next}_1, \ldots, \mathsf{Next}_m, \mathsf{Output})$*:*

- *Next message:* $\mathtt{msg}_i^\ell = \mathsf{Next}_i(1^\lambda, x_i, \rho_i, \overline{\mathtt{msg}}^{<\ell})$ *is the message broadcasted by party $p_i \in \mathcal{P}$ in round $\ell \in [m]$, on input $x_i \in \{0,1\}^\lambda$, on random tape $\rho_i \leftarrow \{0,1\}^\lambda$, after receiving the messages* $\overline{\mathtt{msg}}^{<\ell} = \{\mathtt{msg}_j^{\ell'}\}_{j \in [n], \ell' < \ell}$*, where* $\mathtt{msg}_l^{\ell'}$ *is the message broadcasted by party p_j on round $\ell' \in [\ell - 1]$.*
- *Output:* $y_i = \mathsf{Output}(1^\lambda, x_1, \rho_i, \overline{\mathtt{msg}})$ *is the output of a party p_i for $i = 1, \ldots, n$, on input $x_i \in \{0,1\}^\lambda$, on random tape $\rho_i \in \{0,1\}^\lambda$, after receiving the messages* $\mathtt{msg} = \{\mathtt{msg}_j^\ell\}_{j \in [n], \ell \in [m]}$.

We now recall the notion of security against malicious adversary. We focus on the case with static corruptions and security with abortion. We also recall that we assume that parties have access to a simultaneous broadcast channel. We first define the notions of ideal execution $\mathsf{Ideal}_{I,\mathsf{Sim}}(1^\lambda, \overline{x})$ *against a simulator* Sim *simulating malicious parties $\{P_i\}_{i \in I}$ and of real execution* $\mathsf{Real}_{I,\mathcal{A}}(1^\lambda, \overline{x})$ *against an adversary $\mathcal{A}$ playing the roles of malicious parties $\{P_i\}_{i \in I}$. Simulator* Sim *are defined as non-uniform expected poly-time interactive Turing machines while adversaries $\mathcal{A}$ are defined as non-uniform poly-time interactive Turing machines.*

Ideal Execution. $\mathsf{Ideal}_{I,\mathsf{Sim}}(1^\lambda, \overline{x})$ *is defined by playing the following game with the simulator* Sim*:*

1. *The simulator is given I and $\overline{x}_I$.*

2. *The simulator chooses a vector* $\overline{x}'_I = \{\overline{x}'_i\}_{i \in I}$. *We set* $x'_i = x_i$ *for* $i \in \overline{I}$, *where* $\overline{I} = [N] - I$ *corresponds to the set of honest parties. As usual,* $\overline{x}' = \{\overline{x}'_i\}_{i \in [n]}$.
3. *The simulator is given* $f_I(\overline{x}')$.
4. *The simulator can then decide to abort or proceed. If it aborts, we set* $\overline{y}_{\overline{I}} = (\perp, \ldots, \perp)$, *otherwise, we set* $\overline{y}_{\overline{I}} = f_{\overline{I}}(\overline{x}')$.
5. $\mathsf{Ideal}_{I,\mathsf{Sim}}(1^\lambda, \overline{x})$ *is defined as* $(\overline{y}_{\overline{I}}, z)$ *where* z *is the output of the simulator in the end of the ideal execution.*

Real Execution. $\mathsf{Real}_{I,\mathcal{A}}(1^\lambda, \overline{x})$ *is defined by running the MPC protocol where the adversary* $\mathcal{A}$ *controls the malicious parties* $\{P_i\}_{i \in \overline{I}}$ *while the honest parties* $\{P_i\}_{i \in \overline{I}}$ *follow the protocol. It is then defined as the pair* $(\overline{y}_{\overline{I}}, z)$, *where* $\overline{y}_{\overline{I}}$ *is the vector of outputs of the honest parties while* z *is the output of the adversary. The adversary can be rushing: in each round, it can wait for all the messages from the honest parties before sending its own messages.*

Definition 12 (Maliciously Secure MPC)**.** *Let* n *be a positive integer. Let* f *be an* n*-party functionality. Let* Π *be an MPC protocol for* f. *Then* Π *is secure against malicious adversaries if for any non-uniform poly-time interactive turing machine* $\mathcal{A}$, *there exists a non-uniform expected-poly-time interactive turing machine* $\mathsf{Sim} = \{\mathsf{Sim}_\lambda\}_{\lambda \in \mathbb{N}}$ *such that:*

$$\{\mathsf{Ideal}_{I,\mathsf{Sim}}(1^\lambda, \overline{x})\}_{\lambda, \overline{I}, \overline{x}} \approx \{\mathsf{Real}_{I,\mathcal{A}}(1^\lambda, \overline{x})\}_{\lambda, \overline{I}, \overline{x}}$$

2.7 Delayed-Semi-Malicious

In [HIK+11] the authors introduce the notion of *defensible private protocol* which requires that no adversary participating in the protocol can violate the privacy of an honest party while providing a valid defense. A defense is represented by an input and a random tape that demonstrate that the actions of the adversary are consistent with the description of the protocol. In this work we consider the variant of defensible security defined in [BL18] called *delayed-semi-malicious* security. This requires the adversary to provide a valid defense in the second last round of the protocol, and only in the case that the adversary provides a valid defense the security of the protocol holds. We consider simulation-based security against these adversaries with a universal simulator that can simulate the view of the adversaries by interacting them as black-box in a straight-line way in the same spirit of [BL18]. More formally, we define $\mathsf{Real}^{\mathsf{DSM}}{}_{I,\mathcal{A}}(1^\lambda, \overline{x})$ as $\mathsf{Real}_{I,\mathcal{A}}(1^\lambda, \overline{x})$ except that if at some round $\ell > m - 2$ the adversary outputs an invalid defense, then the honest parties all abort (do not send any more messages) and output $\perp$. The ideal world is defined identically as the ideal world for malicious-secure MPC, except that the simulator Sim, on input $(1^\lambda, I)$ interacts with the adversary $\mathcal{A}$ (as a black bock) in a straight line manner, and receives a defense that $\mathcal{A}$ outputs after round $m - 1$. We denote with $\mathsf{Ideal}_{I,\mathsf{Sim}\leftrightarrow\mathcal{A}}(1^\lambda, \overline{x})$ the output of the honest players and Sim.

Definition 13 (Delayed-Semi-Malicious MPC [BL18])**.** *Let* n *be a positive integer. Let* f *be an* n*-party functionality. Let* Π *be an MPC protocol for* f. *Then* Π *is delayed-semi-maliciously secure if there exists a non-uniform expected-poly-time interactive Turing machine* Sim, *such that, for every non-uniform poly-time interactive Turing machine* $\mathcal{A}$*:*

$$\{\mathsf{Ideal}_{I,\mathsf{Sim}\leftrightarrow\mathcal{A}}(1^\lambda, \overline{x})\}_{\lambda, \overline{I}, \overline{x}} \approx \{\mathsf{Real}^{\mathsf{DSM}}{}_{I,\mathcal{A}}(1^\lambda, \overline{x})\}_{\lambda, \overline{I}, \overline{x}}$$

In [BL18] it is showed how promote a k-round private OT protocol to a delayed-semi-malicious protocol for any $k > 1$.[7]

[7] In [BL18] the authors actually prove a stronger statement showing how to obtain a delayed-semi-malicious MPC protocol from an OT protocol that is secure under an even weaker notion than the one considered in this work.

Theorem 1 (Sec. 9.2 of [BL18])**.** *Assuming that a k-round private oblivious transfer exists then it is possible to obtain a k-round delayed-semi-malicious MPC protocol for any $k > 1$.*

3 Private Oblivious Transfer: Our Construction

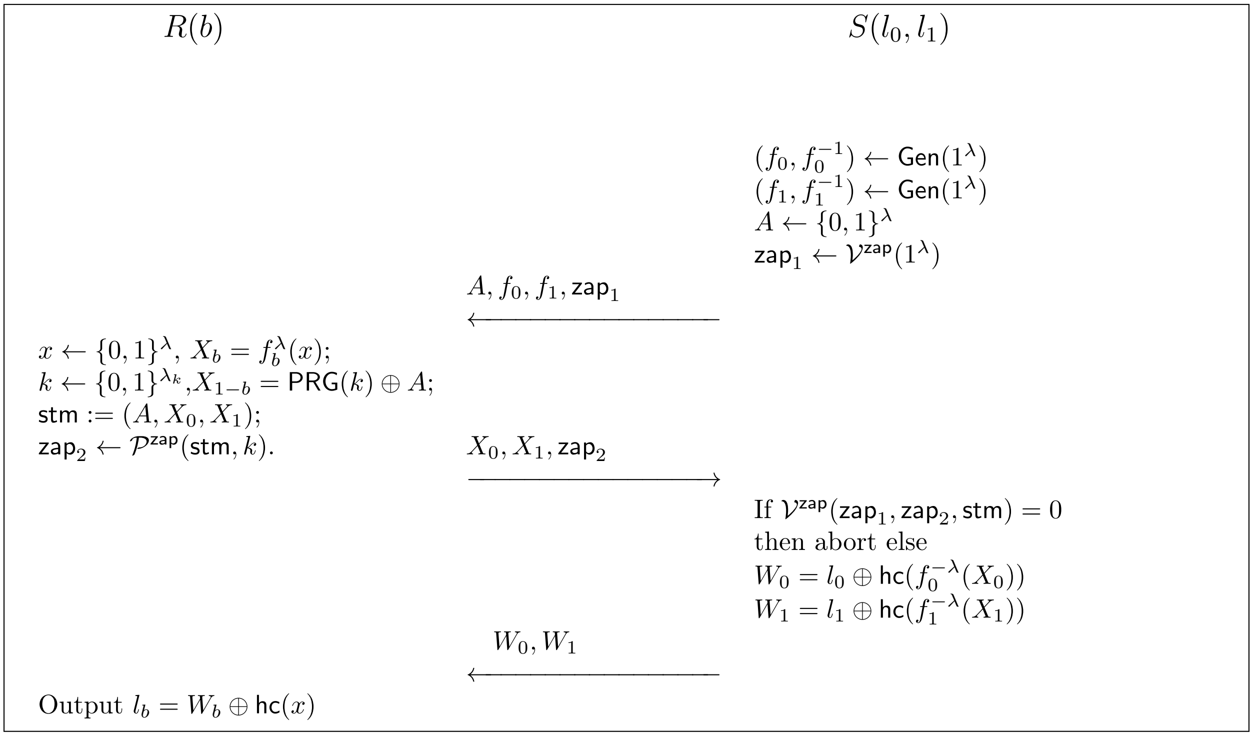
In this section we show how to construct our private 3-round OT assuming sub-exponentially secure TDPs and zap. For our construction we use the following tools.

1. A one-to-one pseudorandom generator $\mathsf{PRG} : \{0,1\}^{\lambda_k} \rightarrow \{0,1\}^{\lambda}$.
2. A two-round resettable witness-indistinguishable proof system $\mathsf{ZAP} = (\mathcal{P}^{\mathsf{zap}}, \mathcal{V}^{\mathsf{zap}})$ for the following $\mathcal{NP}$-language

$$L_{\mathsf{PRG}} = \{(A, X_0, X_1) : \exists k \text{ with } b \in \{0,1\} \text{ s.t. } X_b = \mathsf{PRG}(k) \oplus A\}$$

3. A trapdoor permutation $\mathcal{F} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ with the hardcore bit function $\mathsf{hc}(\cdot)$. We require that no PPT adversary can distinguish the output of $\mathsf{hc}(\cdot)$ from a random value with probability larger than $\frac{1}{2} + \frac{1}{2^{\hat{\lambda}}}$ where $\hat{\lambda} = \lambda/c'$ for some constant c'.

In Fig. 1 we provide the formal description of our 3-round private OT $\Pi^{\mathcal{OT}} = (S, R)$. We let λ be the security parameter and define $\lambda_k = \hat{\lambda}/c$ for some constant $c \geq 2$.



$R(b)$ $\qquad$ $S(l_0, l_1)$

S: $(f_0, f_0^{-1}) \leftarrow \mathsf{Gen}(1^{\lambda})$
$(f_1, f_1^{-1}) \leftarrow \mathsf{Gen}(1^{\lambda})$
$A \leftarrow \{0,1\}^{\lambda}$
$\mathsf{zap}_1 \leftarrow \mathcal{V}^{\mathsf{zap}}(1^{\lambda})$

$\xleftarrow{A, f_0, f_1, \mathsf{zap}_1}$

R: $x \leftarrow \{0,1\}^{\lambda}$, $X_b = f_b^{\lambda}(x)$;
$k \leftarrow \{0,1\}^{\lambda_k}$,$X_{1-b} = \mathsf{PRG}(k) \oplus A$;
$\mathsf{stm} := (A, X_0, X_1)$;
$\mathsf{zap}_2 \leftarrow \mathcal{P}^{\mathsf{zap}}(\mathsf{stm}, k)$.

$\xrightarrow{X_0, X_1, \mathsf{zap}_2}$

S: If $\mathcal{V}^{\mathsf{zap}}(\mathsf{zap}_1, \mathsf{zap}_2, \mathsf{stm}) = 0$
then abort else
$W_0 = l_0 \oplus \mathsf{hc}(f_0^{-\lambda}(X_0))$
$W_1 = l_1 \oplus \mathsf{hc}(f_1^{-\lambda}(X_1))$

$\xleftarrow{W_0, W_1}$

R: Output $l_b = W_b \oplus \mathsf{hc}(x)$

Figure 1: Formal description of our private OT protocol $\Pi^{\mathcal{OT}}$.

Theorem 2. *If sub-exponentially secure one-way trapdoor permutations exist then $\Pi^{\mathcal{OT}}$ is a private oblivious transfert protocol.*

Proof. We first prove that $\Pi^{\mathcal{OT}}$ is secure against malicious receiver and then we prove its security against malicious sender.

Privacy against a malicious receiver. By contradiction there exist: a malicious receiver $R^\star$, a PPT distinguisher D, a non-negligible probability p_{ot} a tuple (l_0, l_1) and z_1 such that with non-negligible probability p_{ot} over $((m_1^s, m_r), \sigma, r_R)$ it holds for $b = 0$ and $b = 1$ that

$$|\text{Prob}\left[\ m_2^s \leftarrow S_2(\sigma, m_r, (l_0, l_1)) : D(1^\lambda, z, r_R, (m_1^s, m_2^s)) = 1\ \right] -$$

$$\text{Prob}\left[\ l_b^\star \leftarrow l_b; l_{1-b}^\star \leftarrow \{0,1\}^\lambda; m_2^s \leftarrow S_2(\sigma, m_r, (l_0^\star, l_1^\star)) : D(1^\lambda, z, r_R, (m_1^s, m_2^s)) = 1\ \right] | \geq p_{ot}.$$

Given the above, we construct and adversary $\mathcal{A}$ that distinguishes between the $h = \mathsf{hc}(x)$ and a random string of λ bits having as input $y = f^\lambda(x)$. This adversary works as follows.

1. $\mathcal{A}$, upon receiving the challenge (f, y, h) picks $k \leftarrow \{0,1\}^{\lambda_k}$ sets $f_0 = f, A = \mathsf{PRG}(k) \oplus y$.
2. $\mathcal{A}$ then computes $(f_1, f_1^{-1}) \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{zap}_1 \leftarrow \mathcal{V}^{\mathsf{zap}}(1^\lambda)$ and sends $(A, f_0, f_1, \mathsf{zap}_1)$ to $R^\star$
3. If $X_0 \neq y$, or in the case that $R^\star$ aborts then $\mathcal{A}$ outputs a random bit. Otherwise, he computes $W_0 = l_0 \oplus h$, $W_1 = l_1 \oplus \mathsf{hc}(f_1^{-\lambda}(X_1))$ and sends (W_0, W_1) to $R^\star$.
4. $\mathcal{A}$ outputs what $R^\star$ outputs.

We observe that in the case that the bit used by $R^\star$ does not correspond to 1, then $\mathcal{A}$ breaks the security of the TDP with probability 1/2. If the bit used by $R^\star$ is 1 and $X_0 = y$, then:

– if h is a random string then the output of $\mathcal{A}$ corresponds to the output of $R^\star$ when S uses as input $(l_0^\star, l_1)$ for some random $l_0^\star$,

– if $h = \mathsf{hc}(x)$ then to the output of $\mathcal{A}$ corresponds to the output of $R^\star$ when S uses (l_0, l_1).

Let us now see formally what is the advantage of $\mathcal{A}$ in breaking the security of $\mathcal{F}$. Without loss of generality, we assume that when $R^\star$ uses the bit 1 as input the probability that $X_0 = y$ is at least $p_1 = {\lambda_k}^{-1}$. Note that we can assume that $X_0 = y$ is at least p_1 since the key k and the challenge y are randomly chosen and the PRG that we are using is one-to-one. Let $p_0 = 1 - p_1$. From the argument given above, the probability that $\mathcal{A}$ has of breaking the security of the TDP is

$$\frac{p_0}{4} + \frac{p_1}{2}(\frac{1}{2} + p_{ot}) = \frac{p_0 + p_1}{2} + \frac{p_1 p_{ot}}{2} = \frac{1}{2} + \frac{p_{ot}}{2^{\lambda_k+1}} \overset{p_{ot} > \frac{1}{2^{\lambda_k - 1}}}{>} \frac{1}{2} + \frac{1}{2^{2\lambda_k}} \overset{\lambda_k = \hat{\lambda}/c}{=} \frac{1}{2} + \frac{1}{2^{\frac{2\hat{\lambda}}{c}}}$$

which for $c \geq 2$ contradicts the security of $\mathcal{F}$ given that we have chosen a one-trapdoor permutation that cannot be broken with probability larger than $\frac{1}{2} + \frac{1}{2^{\hat{\lambda}}}$

□

Privacy against a malicious receiver. By contradiction there exists a PPT adversary $S^\star$ such that the following two ensemble are distinguishable.

$$\{View_{S^\star}[\langle S^\star(1^\lambda), R(1^\lambda, 0)\rangle]\} \quad \{View_{S^\star}[\langle S^\star(1^\lambda), R(1^\lambda, 1)\rangle]\}$$

The proof goes trough hybrid arguments starting from the execution in which the receiver uses the bit 1 as input. We gradually modify this execution until the input of the honest receiver becomes 0. We denote by $\{View_{S^\star}[\mathcal{H}_i]\}$ the output view of $S^\star$ in the hybrid experiment $\mathcal{H}_i$.

-$\mathcal{H}_1$ is identical to the experiment in which the honest receiver interacts with $S^\star$ using 1 as input.

-$\mathcal{H}_2$ differs from $\mathcal{H}_1$ in the way X_1 is computed. In this case $X_1 = \mathsf{PRG}(k') \oplus A$ with $k' \leftarrow \{0,1\}^{\lambda_k}$. We observe that in $\mathcal{H}_1$ X_1 was a uniformly distributed value in $\{0,1\}^\lambda$, given that x is randomly chosen by the receiver and that $X_1 = f_1^\lambda(x)$. The indistinguishability between the two hybrid experiments comes immediately from the security of the PRG.

-$\mathcal{H}_3$ differs from $\mathcal{H}_2$ in the witness used to compute the ZAP proof. More precisely, in $\mathcal{H}_2$ the witness corresponds to the the input of the PRG k such that $X_0 = \mathsf{PRG}(k) \oplus A$ whereas the witness used in $\mathcal{H}_3$ corresponds to the key k' such that $X_1 = \mathsf{PRG}(k') \oplus A$. The indistinguishability between the two hybrid experiments comes from the WI property of ZAP.

-$\mathcal{H}_4$ is equal to $\mathcal{H}_3$ with the difference that $X_1 = f_1^\lambda(x)$ with $x \leftarrow \{0,1\}^\lambda$. The indistinguishability between $\mathcal{H}_4$ and $\mathcal{H}_3$ comes from the security of the PRG for the same arguments used to prove that $\mathcal{H}_1$ and $\mathcal{H}_2$ are indistinguishable. We observe that a reduction to the security of the PRG is possible because the witness used to compute the ZAP proof involves only k' that is such that $X_1 = \mathsf{PRG}(k') \oplus A$. The proof ends with the observation that $\mathcal{H}_4$ corresponds to the execution in which R interacts with $S^\star$ using 0 as input, thus reaching a contradiction. □

4 Proof via Look-Ahead Rewinds

In this section we generalize a proof technique that is required in the security proof of our MPC protocol proposed in the Sec. 5. Consider the scenario in which the adversary $\mathcal{A}$ interacts with two PPT interactive Turing machines (ITM) Ext and B using only four rounds for the communication. The adversary $\mathcal{A}$ interacts during the first three rounds with a PPT extractor Ext using some secret information sk which is fully specified in the third round. The extractor Ext rewinds the adversary from the third to the second round in order to extract sk from $\mathcal{A}$ and in the end of the extraction process Ext goes back to the main thread and outputs the outcome of the extraction.

In addition, from the second to the fourth round $\mathcal{A}$ interacts also with a PPT ITM $\mathsf{B}(y)$ (where y is the input used by B). The Fig. 2 depicts the message-flow that we have just described. Note that since the messages of Ext interleave with B, then the first two messages of B are rewound during the extraction process.

We denote with $\mathsf{Exp}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}(y, z)$ the random variable consisting of the view of $\mathcal{A}(z)$ in an execution when communicating with $\mathsf{B}(y)$ in the latest three rounds and with Ext in the first three rounds as we have just described.

We now consider the execution in which during the look-ahead rewinds, B uses the input y^0, but post rewinds she uses the extracted value $y^1 = \mathsf{sk}$. We denote with $\mathsf{Exp}_{\mathsf{Rew}}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}(y^1, z)$ the random variable consisting of the view of $\mathcal{A}(z)$ in such an execution.

We prove that if for every two sequences $\{y^0_\lambda\}_{\lambda\in\mathbb{N}}$, $\{y^1_\lambda\}_{\lambda\in\mathbb{N}}$ with $y^0_\lambda \in \{0,1\}^\lambda$, $y^1_\lambda \in \{0,1\}^\lambda$, such that for all PPT ITM adversary $\tilde{A}$ it holds that

$$\{\langle \mathsf{B}(y^0_\lambda), \tilde{\mathcal{A}}\rangle(1^\lambda)\}_{\lambda\in\mathbb{N}} \approx \{\langle \mathsf{B}(y^1_\lambda), \tilde{\mathcal{A}}\rangle(1^\lambda)\}_{\lambda\in\mathbb{N}} \tag{1}$$

then it also holds that, for every PPT $\mathcal{A}$,

$$\{\mathsf{Exp}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}(y^0_\lambda, z)\}_{\lambda\in\mathbb{N}} \approx \{\mathsf{Exp}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}_{\mathsf{Rew}}(y^1_\lambda, z)\}_{\lambda\in\mathbb{N}}$$

Roughly, we want to prove that, even in the case that the first and the second round of B are rewound (in order to extract some secret from the adversary that runs also B) the adversary cannot distinguish between when B runs using y_0 and when B uses an input extracted via the look-ahead rewinds ($y_1 = \mathsf{sk}$) under the condition that $\{\langle \mathsf{B}(y^0_\lambda), \tilde{\mathcal{A}}\rangle(1^\lambda)\}_{\lambda\in\mathbb{N}} \approx \{\langle \mathsf{B}(y^1_\lambda = \mathsf{sk}), \tilde{\mathcal{A}}\rangle(1^\lambda)\}_{\lambda\in\mathbb{N}}$.

We note that the proof technique that we are describing here is similar to the one used [HHPV18]. Here we propose a generalization of their technique in order to show that, when the messages of multiple interactive protocols interleave in a specific way, then the look-ahead rewinds do not hurt the security reduction to a primitive (B in our case) involved in the protocol.

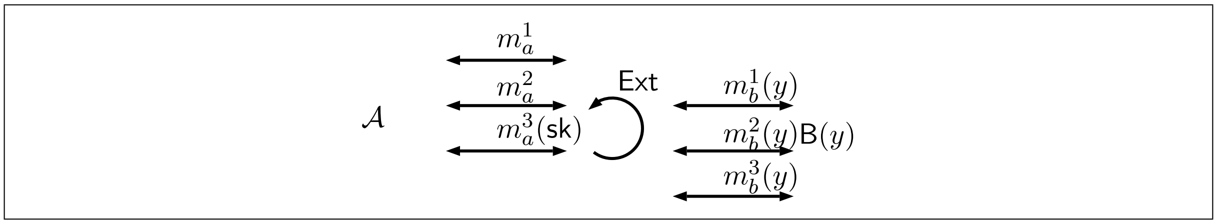


Figure 2: The ITMs communicate via a simultaneous message channel. The first two messages of B interleave with the latest two messages of Ext.

Assume now by contradiction that the claim we are proving is false, then we show how to construct an adversary $\tilde{\mathcal{A}}$ that contradicts the fact that $\{\langle \mathsf{B}(y^0_\lambda), \tilde{\mathcal{A}}\rangle(1^\lambda)\}_{\lambda\in\mathbb{N}} \approx \{\langle \mathsf{B}(y^1_\lambda), \tilde{\mathcal{A}}\rangle(1^\lambda)_{\lambda\in\mathbb{N}}$. $\tilde{\mathcal{A}}$ works as follows. We assume that the security of B can be described as a classical game between a challenger and an attacker, where the challenger takes as input y^0 and y^1, picks $b \in \{0,1\}$ and uses y^b to run B with the adversary$\tilde{\mathcal{A}}$. The adversary $\tilde{\mathcal{A}}$ executes the following steps.

1. Interact with $\mathcal{A}$ by running Ext and $\mathsf{B}(y_0)$ during the look-ahead rewinds.

2. When Ext finishes the extraction then go back to the main thread and act as a proxy between the the external challenger of B and $\mathcal{A}$ for the messages of B until the second round (the third round of the entire protocol).

3. Output what $\mathcal{A}$ outputs.

One crucial observation is that the look-ahead rewinds have no impact on the reduction since the challenger is not even involved during those rewinds. The proof ends with the observation that if the challenger has used y_0 then the output of the experiment corresponds to $\mathsf{Exp}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}(y^0, z)$, to $\mathsf{Exp}^{\mathcal{A},\mathsf{Ext},\mathsf{B}}_{\mathsf{Rew}}(\mathsf{sk}, z)$ otherwise.

5 Round Optimal MPC from Generic Assumptions

In this section we propose a 4-round MPC protocol Π^{MPC}. Our construction makes use of the following tools:

- 3-round delayed-semi-malicious MPC $\mathsf{MPC}^{\mathsf{DSM}} = (\mathsf{Next}_1, \ldots, \mathsf{Next}_3, \mathsf{Output})$
- A one-way permutation $f : \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$ (see App. A for a formal definition of OWP).
- A non-interactive statistically binding commitment scheme $\mathsf{NICOM} = (\mathsf{Com}, \mathsf{Dec})$
- A 3-round special non-malleable commitment scheme $\mathsf{NMCOM} = (\mathsf{Sen}, \mathsf{Rec})$ with honest-extractor $\mathtt{ExtNM}$.
- Let m^ℓ_i be the message broadcast by P_i in round ℓ of $\mathsf{MPC}^{\mathsf{DSM}}$ and $\overline{\mathtt{msg}}^{<\ell} = \{\mathtt{msg}^{\ell'}_i\}_{i\in[n],\ell'<\ell}$ be the messages broadcasted by all the parties in the first $\ell - 1$ rounds of $\mathsf{MPC}^{\mathsf{DSM}}$, then we use a resettable two-round witness-indistinguishable proof systems $\mathsf{ZAP} = (\mathcal{P}^{\mathsf{zap}}, \mathcal{V}^{\mathsf{zap}})$ for the following $\mathcal{NP}$-language

$$\begin{aligned} L^{i\to j}_{\mathsf{ZAP}} = \Big\{\big(\overline{\mathtt{msg}}^{<\,2}, \mathtt{msg}^2_i, (\mathsf{nm}^{i\to j}_{0,1}, \mathsf{nm}^{j\to i}_{0,2}, \mathsf{nm}^{i\to j}_{0,3}), (\mathsf{nm}^{i\to j}_{1,1}, \mathsf{nm}^{j\to i}_{1,2}, \mathsf{nm}^{i\to j}_{1,3}), Y^{i\to j}_0, Y^{i\to j}_1\big) :& \\ \exists(\mathsf{dec}_{\mathsf{nm}}, y) \text{ s.t. } (Y^{i\to j}_b = f(y) \text{ with } b \in \{0,1\} \text{ AND}& \\ \mathsf{Rec}(\mathsf{nm}^{i\to j}_{1,1}, \mathsf{nm}^{j\to i}_{1,2}, \mathsf{nm}^{i\to j}_{1,3}, \mathsf{dec}_{\mathsf{nm}}, y) = 1)& \\ \text{OR } (\mathsf{Rec}(\mathsf{nm}^{i\to j}_{0,1}, \mathsf{nm}^{j\to i}_{0,2}, \mathsf{nm}^{i\to j}_{0,3}, \mathsf{dec}_{\mathsf{nm}}, y) = 1 \text{ AND}& \\ \forall \ell \le 2\ m^\ell_i = \mathsf{Next}_\ell(1^\lambda, y, \overline{\mathtt{msg}}^{<\,\ell}))\Big\}.& \end{aligned}$$

  Informally, in our scheme ZAP is used by the party P_i to prove to the party P_j that either a transcript for a special non-malleable commitment scheme contains y such that $y = f(Y^{j\to i}_b)$ where $Y^{j\to i}_b$ was chosen by P_j, or that a the the first two rounds of the DSM MPC protocol $\mathsf{MPC}^{\mathsf{DSM}}$ are computed honestly and that the defence for $\mathsf{MPC}^{\mathsf{DSM}}$ is committed using a special non-malleable commitment scheme.
- A Σ-protocol $\mathsf{BL}_L = (\mathcal{P}_L, \mathcal{V}_L)$ for the $\mathcal{NP}$-language $L = \{Y : \exists\ y \text{ s.t. } Y = f(y)\}$ with Special HVZK simulator Sim_L. We uses two instantiations of BL_L in order to construct the protocol Π_{OR} for the OR of two statements following the approach proposed in [COSV17] (see Sec. A.2 for an overview of this technique). Π_{OR} is a proof of knowledge for the $\mathcal{NP}$-language $L_{\mathsf{OR}} = \{(Y_0, Y_1) : \exists\ y \text{ s.t. } f(y) = Y_0 \text{ OR } f(y) = Y_1\}$ [8]. Informally, by running Π_{OR}, one can prove the knowledge of either the preimage of Y_0 or Y_1 given that f is a OWP.
- A 4-round delayed-input synchronous many-many NMZK AoK $\mathsf{NMZK} = (\mathcal{P}^{\mathsf{ZK}}, \mathcal{V}^{\mathsf{ZK}})$ for the following $\mathcal{NP}$-language

$$\begin{aligned} L^{i\to j}_{\mathsf{NMZK}} = \Big\{\big(\overline{\mathtt{msg}}^{<\,3}, \mathtt{msg}^3_i, (\mathsf{nm}^{i\to j}_1, \mathsf{nm}^{j\to i}_2, \mathsf{nm}^{i\to j}_3), \mathtt{com}\big) :& \\ \exists(\mathsf{dec}_{\mathsf{nm}}, y, x, \mathsf{dec}) \text{ s.t. } \mathsf{Rec}(\mathsf{nm}^{i\to j}_1, \mathsf{nm}^{j\to i}_2, \mathsf{nm}^{i\to j}_3, \mathsf{dec}_{\mathsf{nm}}, (\rho, x)) = 1& \\ \text{AND } \forall \ell \le 3\ m^\ell_i = \mathsf{Next}_\ell(1^\lambda, (\rho, x), \overline{\mathtt{msg}}^{<\,\ell}) \text{ AND } \mathsf{Dec}(\mathtt{com}, x, \mathsf{dec}) = 1\Big\}.& \end{aligned}$$

  Informally, by running NMZK one can prove that a valid defence for all the messages of

[8] We use Π_{OR} in a non-black box way, but for ease of exposition sometimes we will refer to entire protocol Π_{OR} in order to invoke the proof of knowledge property enjoyed by Π_{OR}.

$\mathsf{MPC}^{\mathsf{DSM}}$ is committed via a special non-malleable commitment scheme and the the input used for to run $\mathsf{MPC}^{\mathsf{DSM}}$ is committed using a non-interactive commitment scheme.

5.1 Informal Description

In Fig. 3 we propose an informal description of our protocol. The Figure depicts the messages flow the goes from one party P_i to a party P_j. The overall structure of our protocol can be summarized as as follows. Each party commits to his input in the first round using a non-interactive commitment scheme. The parties engage an DSM MPC protocol $\mathsf{MPC}^{\mathsf{DSM}}$ in the latest three rounds of the protocol and prove that $\mathsf{MPC}^{\mathsf{DSM}}$ has been execute correctly. That is, each party P_i proves that the second round of $\mathsf{MPC}^{\mathsf{DSM}}$ is well formed by committing to its randomness and input (which represent a defence for $\mathsf{MPC}^{\mathsf{DSM}}$) using a special non-malleable commitment scheme, and by proving via ZAP that this commitment actually contains a valid defence. To ensure that also the third round of $\mathsf{MPC}^{\mathsf{DSM}}$ is well formed (and that the defence is consistent with the value committed in the first round) the parties engage a delayed-input NMZK Argument. In addition, in the first three rounds P_j engages a WIPoK with P_i to prove the knowledge a value y such that $Y_b = f(y)$. Moreover, a special non-malleable commitment of a random string is sent from P_i to P_j. The aim of those two components is to help the simulator. Indeed, the simulator will extract y from the WIPoK and will commit to it using the special non-malleable commitment scheme. Then the simulator will use the knowledge of y and the knowledge of the decommitment information as a witness for the ZAP. We note that the simulator, in order to extract y, needs to rewind the adversary from the third to the second round. This means that before the rewinds he has to use a valid witness for the ZAP without knowing y. Our simulator during the look-ahead rewinding threads will use a valid defence for $\mathsf{MPC}^{\mathsf{DSM}}$ with a random input (i.e., a value that is inconsistent with what is committed in the first round via the non-interactive commitment scheme). After the extraction, Sim rewinds up to the second round, commits to the trapdoor, uses the simulator $\mathsf{MPC}^{\mathsf{DSM}}$ and complete the *zap* proof using the knowledge of the trapdoor (and the knowledge of the the decommitment information of the special non-malleable commitment used to commit to the trapdoor).

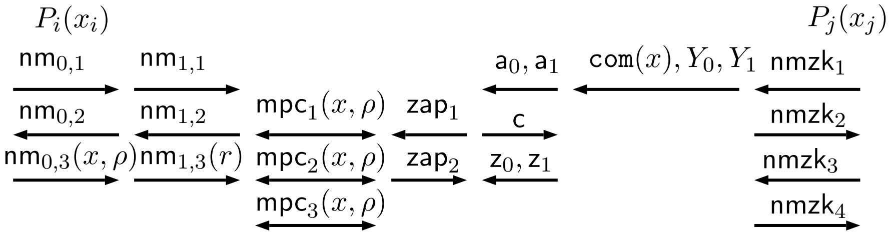
5.2 Formal Description of Π^{MPC}

Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be the set of parties. We describe the messages that P_i computes and sends to P_j in any rounds of Π^{MPC} for all $i, j \in \{1, \dots, n\}$ with $i \neq j$ and $P_i, P_j \in \mathcal{P}$. We denote a message x that goes from P_i to P_j with $x^{i \to j}$ and denote with $\overline{\mathtt{msg}}^{<i}$ all the messages of $\mathsf{MPC}^{\mathsf{DSM}}$ that have been sent by all the parties up to the $(i-1)$-th round of $\mathsf{MPC}^{\mathsf{DSM}}$.

Protocol Π^{MPC}. *Common input:* security parameter λ, instances length: ℓ_{ZAP}, ℓ_{NMZK}. *Private input of party P_i: x_i*

Round 1. For all $j \in \{1, \dots, n\} \setminus \{i\}$ P_i executes the following steps.

1. Compute $y_0^{i \to j} \leftarrow \{0,1\}^\lambda$, $Y_0^{i \to j} \leftarrow f(y_0^{i \to j})$ and pick $Y_1^{i \to j} \leftarrow \{0,1\}^\lambda$.
2. Compute $\mathsf{a}_0^{i \to j} \leftarrow \mathcal{P}_L(1^\lambda, Y_0^{i \to j}, y_0^{i \to j})$.
3. Pick $\mathsf{c}_1^{i \to j} \leftarrow \{0,1\}^\lambda$ and compute $(\mathsf{a}_1^{i \to j}, \mathsf{z}_1^{i \to j}) \leftarrow \mathsf{Sim}_L(1^\lambda, Y_1^{i \to j}, \mathsf{c}_1)$.
4. Pick $r \leftarrow \{0,1\}^\lambda$ and run Sen on input 1^λ and r thus obtaining $(\mathsf{nm}_{0,1}^{i \to j}, \mathsf{dnm}_0)$ (where dnm_0 represents the decommitment information).
5. Pick $r' \leftarrow \{0,1\}^\lambda$ and run Sen on input 1^λ and r' thus obtaining $(\mathsf{nm}_{1,1}^{i \to j}, \mathsf{dnm}_1)$.



- $(\mathsf{mpc}_1, \mathsf{mpc}_2, \mathsf{mpc}_3)$ represents the transcript of a DSM MPC protocol obtained using x as input and ρ as the randomness.
- $(\mathsf{nm}_{0,1}, \mathsf{nm}_{0,2}, \mathsf{nm}_{0,3})$ is the transcript of the special non-malleable commitment scheme that commits to the input and the randomness used to compute the messages of the DSM MPC protocol.
- $(\mathsf{nm}_{1,1}, \mathsf{nm}_{1,2}, \mathsf{nm}_{1,3})$ is the transcript of the special non-malleable commitment scheme that it is used to commit to a random message r. The simulator will commit to the pre-image of either Y_0 or Y_1 and use the knowledge of that pre-image, together with the decommitment information of the non-malleable commitment, as a witness for ZAP.
- $(\mathsf{zap}_1, \mathsf{zap}_2)$ are the two rounds of the zap proof the proves either that $(\mathsf{nm}_{0,1}, \mathsf{nm}_{0,2}, \mathsf{nm}_{0,3})$ contains (x, ρ), where (x, ρ) denotes the input and the randomness used to compute the messages of the DSM MPC protocol, or that $(\mathsf{nm}_{1,1}, \mathsf{nm}_{1,2}, \mathsf{nm}_{1,3})$ contains a pre-image of Y_b with $b \in \{0, 1\}$.
- $\mathtt{com}$ is a non-interactive commitment of the input and (Y_0, Y_1) are two elements taken from the domain of a one-way permutation.
- $(\mathsf{a}_0, \mathsf{a}_1, \mathsf{c}_0, \mathsf{c}_1, \mathsf{z}_0, \mathsf{z}_1)$ is the transcript generated from an execution of the WIPoK Π_{OR} in which $\mathcal{P}_{\mathsf{OR}}$ proves the knowledge of either the pre-image Y_0 or Y_1.
- $(\mathsf{nmzk}_1, \mathsf{nmzk}_2, \mathsf{nmzk}_3, \mathsf{nmzk}_4)$ in the transcript generated from an execution of the delayed-input synchronous many-many NMZK NMZK in which $\mathcal{P}_{\mathsf{NMZK}}$ proves that: 1) The defense of the DSM MPC protocol (ρ, x) is committed in one of the special non-malleable commitment, 2) all the messages of DSM MPC are computed correctly; 3) $\mathtt{com}$ contains a commitment of x which is compatible with the defense of the DSM MPC protocol.

Figure 3: The flow of messages that goes from the party P_i to a party P_j in Π^{MPC}.

6. Compute $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(1^\lambda, x_i)$
7. Run $\mathcal{V}^{\mathsf{ZK}}$ on input 1^λ thus obtaining $\mathsf{nmzk}_1^{i\to j}$
8. Send $(Y_0^{i\to j}, Y_1^{i\to j}, \mathsf{a}_0^{i\to j}, \mathsf{a}_1^{i\to j}, \mathsf{nm}_{0,1}^{i\to j}, \mathsf{nm}_{1,1}^{i\to j}, \mathtt{com}^{i\to j}, \mathsf{nmzk}_1^{i\to j})$ to P_j.

Round 2. On input $\{Y_0^{j\to i}, Y_0^{j\to i}, \mathsf{a}_0^{j\to i}, \mathsf{a}_1^{j\to i}, \mathsf{nm}_{0,1}^{j\to i}, \mathsf{nm}_{1,1}^{j\to i}, \mathtt{com}^{j\to i}, \mathsf{nmzk}_1^{j\to i}\}_{j\in\{1,\dots,n\}\setminus\{i\}}$, for all $j \in \{1, \dots, n\} \setminus \{i\}$ P_i executes the following steps.

1. Pick $\mathsf{c}^{i\to j} \leftarrow \{0, 1\}^\lambda$.
2. Run Rec on input $\mathsf{nm}_{0,1}^{j\to i}$ thus obtaining $\mathsf{nm}_{0,2}^{i\to j}$.
3. Run Rec on input $\mathsf{nm}_{1,1}^{j\to i}$ thus obtaining $\mathsf{nm}_{1,2}^{i\to j}$.
4. Pick $\rho^{i\to j} \leftarrow \{0, 1\}^\lambda$ and compute $\mathtt{msg}_i^1 \leftarrow \mathsf{Next}_1(1^\lambda, x; \rho^{i\to j})$.
5. Run $\mathcal{P}^{\mathsf{zap}}$ on input 1^λ thus obtaining $\mathsf{zap}_1^{i\to j}$.

6. Run $\mathcal{P}^{\mathsf{ZK}}$ on input $(1^\lambda, \mathsf{nmzk}_1^{j\to i})$ thus obtaining $\mathsf{nmzk}_2^{i\to j}$.
7. Send $(\mathsf{c}^{i\to j}, \mathsf{nm}_{0,2}^{i\to j}, \mathsf{nm}_{1,2}^{i\to j}, \mathtt{msg}_i^1, \mathsf{zap}_1^{i\to j}, \mathsf{nmzk}_2^{i\to j})$ to P_j.

Round 3. On input $\{\mathsf{c}^{j\to i}, \mathsf{nm}_{0,2}^{j\to i}, \mathsf{nm}_{1,2}^{j\to i}, \mathsf{zap}_1^{j\to i}, \mathsf{nmzk}_2^{j\to i}\}_{j\in\{1,\dots,n\}\setminus\{i\}}, \overline{\mathtt{msg}}^{<\,2}$, for all $j \in \{1,\dots,n\} \setminus \{i\}$ P_i executes the following steps.

1. Commit to $(x_i, \rho^{i\to j})$ by running Sen on input $((x_i, \rho^{i\to j}), \mathsf{nm}_{0,1}^{i\to j}, \mathsf{nm}_{0,2}^{j\to i}, \mathsf{dnm}_0^{i\to j})$ thus obtaining $\mathsf{nm}_{0,3}^{i\to j}$ (we recall that the special non-malleable commitment scheme is delayed-input).
2. Pick $r^{i\to j} \leftarrow \{0,1\}^\lambda$ and commit to it by running Sen on input $(r^{i\to j}, \mathsf{nm}_{1,1}^{i\to j}, \mathsf{nm}_{1,2}^{j\to i}, \mathsf{dnm}_1^{i\to j})$ thus obtaining $\mathsf{nm}_{1,3}^{i\to j}$.
3. Set $x_{\mathsf{ZAP}}^{i\to j} = \big(\overline{\mathtt{msg}}^{<3}, (\mathsf{nm}_{0,1}^{i\to j}, \mathsf{nm}_{0,2}^{j\to i}, \mathsf{nm}_{0,3}^{i\to j}), (\mathsf{nm}_{1,1}^{i\to j}, \mathsf{nm}_{1,2}^{j\to i}, \mathsf{nm}_{1,3}^{i\to j}), Y_0^{i\to j}, Y_1^{i\to j}\big)$ and $w_{\mathsf{ZAP}} = (\mathsf{dnm}_0, x_i, \rho)$ and run $\mathcal{P}^{\mathsf{zap}}$ in input $(x_{\mathsf{ZAP}}^{i\to j}, w_{\mathsf{ZAP}}, \mathsf{zap}_1^{j\to i})$ thus obtaining $\mathsf{zap}_2^{i\to j}$.
4. Compute $\mathsf{c}_0^{i\to j} = \mathsf{c}^{j\to i} \oplus \mathsf{c}_1^{i\to j}$ and $\mathsf{z}_0^{i\to j} \leftarrow \mathcal{P}_L(Y_0^{i\to j}, y_0^{i\to j}, \mathsf{c}_0^{j\to i})$.
5. Compute $\mathtt{msg}_i^2 = \mathsf{Next}_i(1^\lambda, x_i, \rho_i, \overline{\mathtt{msg}}^{<\,2}\,)$
6. Run $\mathcal{V}^{\mathsf{ZK}}$ on input $\mathsf{nmzk}_2^{j\to i}$ thus obtaining $\mathsf{nmzk}_3^{i\to j}$.
7. Send $(\mathsf{nm}_{0,3}^{i\to j}, \mathsf{nm}_{1,3}^{i\to j}, \mathsf{zap}_2^{i\to j}, (\mathsf{z}_0^{i\to j}, \mathsf{c}_0^{i\to j}), (\mathsf{z}_1^{i\to j}, \mathsf{c}_1^{i\to j}), \mathtt{msg}_i^2, \mathsf{nmzk}_3^{i\to j})$ to P_j

Round 4. On input $\{\mathsf{nm}_{0,3}^{j\to i}, \mathsf{nm}_{1,3}^{j\to i}, (\mathsf{zap}_2^{j\to i}, x_{\mathsf{ZAP}}^{j\to i}), (\mathsf{z}_0^{j\to i}, \mathsf{c}_0^{j\to i}), (\mathsf{z}_1^{j\to i}, \mathsf{c}_1^{j\to i}), \mathsf{nmzk}_3^{j\to i}\}_{j\in\{1,\dots,n\}\setminus\{i\}}, \overline{\mathtt{msg}}^{<\,3}$ for all $j \in \{1,\dots,n\} \setminus \{i\}$ P_i executes the following steps.

1. Check that the following conditions are satisfied:
   - $x_{\mathsf{ZAP}}^{j\to i}$ is constructed accordingly to the protocol description;
   - $\mathsf{c}^{i\to j} = \mathsf{c}_0^{j\to i} \oplus \mathsf{c}_1^{j\to i}$;
   - $\mathcal{V}_L(\mathsf{a}_0^{j\to i}, \mathsf{c}_0^{j\to i}, \mathsf{z}_0^{j\to i}, Y_0^{j\to i}) = 1$;
   - $\mathcal{V}_L(\mathsf{a}_1^{j\to i}, \mathsf{c}_1^{j\to i}, \mathsf{z}_1^{j\to i}, Y_1^{j\to i}) = 1$;
   - $\mathcal{V}^{\mathsf{zap}}(\mathsf{zap}_1^{i\to j}, \mathsf{zap}_2^{j\to i}, x_{\mathsf{ZAP}}^{j\to i}) = 1$.
2. Compute $\mathtt{msg}_i^3 = \mathsf{Next}_i(1^\lambda, x_i, \rho_i, \overline{\mathtt{msg}}^{<\,3}\,)$.
3. Set $x_{\mathsf{NMZK}}^{i\to j} = \big(\overline{\mathtt{msg}}^{<\,3}\,, \mathtt{msg}_i^3, (\mathsf{nm}_{0,1}^{i\to j}, \mathsf{nm}_{0,2}^{j\to i}, \mathsf{nm}_{0,3}^{i\to j}), \mathtt{com}\big)$ and $w_{\mathsf{NMZK}} = (\mathsf{dnm}_0^{i\to j}, x_i, \rho^{i\to j}, \mathtt{dec})$ and run $\mathcal{P}^{\mathsf{ZK}}$ on input $(x_{\mathsf{NMZK}}^{i\to j}, w_{\mathsf{NMZK}}, \mathsf{nmzk}_3^{j\to i})$ thus obtaining $\mathsf{nmzk}_4^{i\to j}$.
4. Send $(\mathtt{msg}_i^3, x_{\mathsf{NMZK}}^{i\to j}, \mathsf{NMZK}_4^{i\to j})$ to P_j.

Output computation. On input $\{x_{\mathsf{NMZK}}^{j\to i}, \mathsf{NMZK}_4^{j\to i})\}_{j\in\{1,\dots,n\}\setminus\{i\}}, \overline{\mathtt{msg}}^{<\,4}$, for all $j \in \{1,\dots,n\} \setminus \{i\}$ P_i checks if the following conditions are satisfied:

1. $x_{\mathsf{NMZK}}^{j\to i}$ is constructed according to the protocol description;
2. $\mathcal{V}^{\mathsf{ZK}}$ on input $(\mathsf{NMZK}_4^{j\to i}, x_{\mathsf{NMZK}}^{j\to i})$ outputs 1.

If the above conditions are satisfied then P_i compute $y_i = \mathsf{Output}(1^\lambda, x_1, \rho_i, \overline{\mathtt{msg}}^{<\,4}\,)$ and output y_i.

Our simulator: high level description. Before formally prove the security of our protocol, we give more details on how the simulator $\mathsf{Sim}_{\mathsf{MPC}}$ works (we refer the reader to Fig. 4 for the formal description of $\mathsf{Sim}_{\mathsf{MPC}}$). $\mathsf{Sim}_{\mathsf{MPC}}$ works in two phases. In the first phase $\mathsf{Sim}_{\mathsf{MPC}}$ runs internally the adversary $\mathcal{A}$ up to the third round by running the simulator for NMZK and by computing all the other messages of Π^{MPC} as the honest parties would do. That is, $\mathsf{Sim}_{\mathsf{MPC}}$ runs $\mathsf{MPC}^{\mathsf{DSM}}$ using a random input x, commits to the defense of $\mathsf{MPC}^{\mathsf{DSM}}$ and proves via ZAP that the defence is committed via the special non-malleable commitment scheme. $\mathsf{Sim}_{\mathsf{MPC}}$ now run the extractor of the special non-malleable commitment scheme and the PoK extractor of Π_{OR} thus rewinding from the third to the second round.

In the end of the extraction process the PoK extractor outputs the trapdoors (which correspond to the inverses of the OWPs pre-images sent from the parties controlled from the adversary), and the extractor of the special non-malleable commitment scheme outputs the decommitment information d. $\mathsf{Sim}_{\mathsf{MPC}}$ now rewinds $\mathcal{A}$ up to the second round, and interacts with $\mathcal{A}$ using the simulator $\mathsf{Sim}_{\mathsf{DSM}}$ of $\mathsf{MPC}^{\mathsf{DSM}}$. Moreover, in order to compute an accepting proof for the ZAP, $\mathsf{Sim}_{\mathsf{MPC}}$ commits to the trapdoor via the special non-malleable commitment scheme and use the knowledge of the trapdoor (and of the decommitment information) as a witness for the ZAP. We recall that $\mathsf{Sim}_{\mathsf{DSM}}$, in order to compute the last round, requires a valid defence. This defence is extracted using d due to the fact that the decommitment information d can be reused since the first round of NMCOM is the same both in the main and in the look-ahead threads. The crucial part of the proof is to show that $\mathsf{Sim}_{\mathsf{MPC}}$ is able to extract a valid defence from the non-malleable commitment that is only honest-extractable. Moreover, in the proof it is important to show that the adversarial parties behave honestly with respect to the messages of $\mathsf{MPC}^{\mathsf{DSM}}$ also after the rewinds.

Theorem 3. *If* NMCOM *is a special non-malleable commitment scheme,* $\mathsf{MPC}^{\mathsf{DSM}}$ *is a 3-round delayed-semi-malicious MPC protocol,* f *is a OWP and* ZAP *exits then* Π^{MPC} *is 4-round maliciously secure MPC protocol.*

Proof. To prove our theorem we need to show a non-uniform expected-poly-time simulator $\mathsf{Sim}_{\mathsf{MPC}} = \{\mathsf{Sim}_{\mathsf{MPC}\lambda}\}_{\lambda\in\mathbb{N}}$ such that:

$$\{\mathsf{Ideal}_{I,\mathsf{Sim}_{\mathsf{MPC}}}(1^{\lambda},\overline{x})\}_{\lambda,\overline{I},\overline{x}} \approx \{\mathsf{Real}_{I,\mathcal{A}}(1^{\lambda},\overline{x})\}_{\lambda,\overline{I},\overline{x}}$$

We propose the description of $\mathsf{Sim}_{\mathsf{MPC}}$ in Fig. 4. Our proof proceeds via hybrid arguments. Without loss of generality, in the proof we assume that there is only one honest party P_i.

-$\mathcal{H}_0$. This hybrid experiment corresponds to the standard execution of Π^{MPC} where P_i interacts with the malicious party according to the description of Π^{MPC}. We assume that $\mathcal{H}_0$ ends successfully (P_i does not abort) with non-negligible probability (otherwise the theorem is trivially proven). We denote with $\mathsf{NOTRAP}_{\mathcal{H}_i}$ the event in which the adversary provides a well-formed non-malleable commitment of y such that $Y_b^{i\to j}$ for some $j \in \{1,\ldots,n\} - \{i\}$ and $b \in \{0,1\}$ in the hybrid experiment $\mathcal{H}_i$. We want to prove the following lemma

Lemma 1. $\mathrm{Prob}\,[\,\mathsf{NOTRAP}_{\mathcal{H}_0}\,] < \nu(\lambda)$

Proof. Suppose by contradiction that the lemma does not hold. This means that there exists j such that $(\mathsf{nm}_{1,1}^{j\to i}, \mathsf{nm}_{1,2}^{i\to j}, \mathsf{nm}_{1,3}^{j\to i})$ is a well formed non-malleable commitment of a value y with $Y_b^{i\to j} = f(y)$ and $b \in \{0,1\}$. Since NMCOM is honest extractable and the commitment is well formed, it is possible to extract y via rewinds. We now observe that if $Y_1^{i\to j} = f(y)$ we can break the one-way property of f because y is never used by the honest party. That is, $\mathsf{a}_0^{i\to j}$ is computed using the witness y_0 s.t. $Y_0 = f(y_0)$. However, if $Y_0 = f(y)$ then the reduction is not immediate since y is actually used in the hybrid experiment. In order to reach a contradiction, we consider the intermediate hybrid experiment $\mathcal{H}_0^{y_1}$ in which the witness used in Π_{OR} is y_1 s.t. $Y_1^{i\to j} = f(y_1)$. If we now rewind the adversary and the value extracted from the non-malleable commitment is y such that $Y_0 = f(y)$ then a reduction that breaks the one-wayness of f can be done. In any other case we can construct a reduction that breaks the Special HVZK of BL. First, we observe that we cannot simply rely on the WI property of Π_{OR} since the rewinds required to extract from the special non-malleable commitment scheme could rewind the third and the second messages of

Π_{OR} thus interfering with the reduction. For this reason we now consider Π_{OR} in a non-black-box way and consider an additional intermediate hybrid experiment $\mathcal{H}_0^{y_0,y_1}$. $\mathcal{H}_0^{y_0,y_1}$ follows the same steps of $\mathcal{H}_0$ except that the honest prover procedure ($\mathcal{P}_L$), instead of the Special HVZK simulator (Sim_L), is used to compute the prover's messages $\mathsf{a}_1^{i\to j}, \mathsf{z}_1^{i\to j}$ of the transcript $\tau_1 = (\mathsf{a}_1^{i\to j}, \mathsf{c}_1^{i\to j}, \mathsf{z}_1^{i\to j})$ w.r.t. the instance $Y_1^{i\to j}$. Suppose now by contradiction that the value extracted from the special non-malleable commitment scheme is y such that $Y_0^{i\to j} \neq f(y)$, then we can show a malicious verifier $\mathcal{V}^\star$ that distinguishes between a transcript $\tau_1 = (\mathsf{a}_1, \mathsf{c}_1, \mathsf{z}_1)$ computed using Sim_L and one computed using the honest prover procedure. In more details, let $\mathcal{C}_{\mathsf{SHVZK}}$ be the challenger of the Special HVZK. $\mathcal{V}^\star$ picks $\mathsf{c}_1 \leftarrow \{0,1\}^\lambda$ and sends c_1 to $\mathcal{C}_{\mathsf{SHVZK}}$. Upon receiving $\mathsf{a}_1, \mathsf{z}_1$ from $\mathcal{C}_{\mathsf{SHVZK}}$ $\mathcal{V}^\star$ interacts with the adversary according to $\mathcal{H}_0$ ($\mathcal{H}_0^{y_0,y_1}$) except for the messages of τ_1 where $\mathcal{V}^\star$ acts as a proxy between $\mathcal{C}_{\mathsf{SHVZK}}$ and $\mathcal{A}$. At the end of the execution $\mathcal{V}^\star$ runs the extractor of the special non-malleable commitment scheme thus obtaining $\tilde{y}$. Therefore, we have reached a contradiction. We observe that if $\mathcal{C}_{\mathsf{SHVZK}}$ sends a simulated transcript then $\tilde{y} = y_0$ otherwise $\tilde{y} \neq y_0$.

There is a subtlety in the above reduction. $\mathcal{V}^\star$ runs the extractor for the special non-malleable commitment scheme that rewinds from the third to the second round. This means that $\mathcal{V}^\star$ has to be able to complete during the rewinds the third round while receiving different challenges $\mathsf{c}^1, \dots, \mathsf{c}^{\mathsf{poly}(\lambda)}$ w.r.t. Π_{OR}. Since we are splitting the challenge c, $\mathcal{V}^\star$ can just keep fixed the value c_1 reusing the same z_1 (sent by $\mathcal{C}_{\mathsf{SHVZK}}$) and computing an answer to a_0 using the knowledge of y_0 s.t. $Y_0^{i\to j} = f(y_0)$. We note now that $\mathcal{H}_0^{y_1}$ proceeds exactly as $\mathcal{H}_0^{y_0,y_1}$ except that the Special HVZK simulator (Sim_L), instead of honest procedure ($\mathcal{P}_L$), is used to compute the prover's messages $\mathsf{a}_0^{i\to j}, \mathsf{z}_0^{i\to j}$ of the transcript $\tau_0^{i\to j} = (\mathsf{a}_0^{i\to j}, \mathsf{c}_0^{i\to j}, \mathsf{z}_0^{i\to j})$ w.r.t. the instance y_0. From the same arguments proposed above, we can prove that the value extracted from the special non-malleable commitment scheme is y_0 s.t. $Y_0^{i\to j} = f(y_0)$ also in $\mathcal{H}_0^{y_1}$. We note that this is sufficient to reach a contradiction since y_0 is not used in the hybrid experiment. That is, the adversary can be used to break the security of the OWP. □

Another property that we need to show to hold in this (and in the next hybrid experiments) is that the adversary behaves correctly. More formally, we require that if P_j does not abort, then $x_{\mathsf{NMZK}}^{j\to i}$ is a true statement for all $j \in \{1, \dots, n\} - \{i\}$. We denote the event in which there exists $j \in \{1, \dots, n\} - \{i\}$ such that P_j does not abort with non-negligible probability but $x_{\mathsf{NMZK}}^{j\to i}$ is a false statement in the hybrid $\mathcal{H}_i$ with $\mathsf{FALSEX}_{\mathcal{H}_i}$. Due to the soundness of NMZK we can claim that $\text{Prob}\left[\, \mathsf{FALSEX}_{\mathcal{H}_0} \,\right] < \nu(\lambda)$.

–$\mathcal{H}_1$ is equal to $\mathcal{H}_0$ with the difference that the simulator of NMZK is run instead of the honest prover procedure. The indistinguishability between $\mathcal{H}_0$ and $\mathcal{H}_1$ comes immediately from the Zero-knowledge of NMZK. Moreover, from the many-many non-malleability of NMZK we can claim that the adversary is still proving true theorems, that is $\text{Prob}\left[\, \mathsf{FALSEX}_{\mathcal{H}_1} \,\right] < \nu(\lambda)$. We observe that $\text{Prob}\left[\, \mathsf{NOTRAP}_{\mathcal{H}_1} \,\right] < \nu(\lambda)$ since otherwise a reduction that breaks the Zero-Knowledge of NMZK can be made[9].

–$\mathcal{H}_2$ is equal to $\mathcal{H}_1$ with the difference that it rewinds the adversary from the third to the second round in order to get the witness used by the adversary in Π_{OR}. $\mathcal{H}_1$ and $\mathcal{H}_2$ are statistically indistinguishable due to the PoK property of Π_{OR}, moreover $\text{Prob}\left[\, \mathsf{FALSEX}_{\mathcal{H}_2} \,\right] < \nu(\lambda)$ holds otherwise a reduction to the PoK property of Π_{OR} can be made. The reduction would construct a malicious

[9] Note that we are assuming that the simulator of NMZK rewinds from the third to the second round, and from the fourth to the third. This means that the extraction from the special non-malleable commitment scheme can be done concurrently with the rewinds made by the simulator of NMCOM.

prover for Π_{OR} by running internally the adversary for Π^{MPC} and by starting an interaction with the challenger of PoK. In the end of the interaction between the challenger and the malicious prover, the reduction can check the output of the simulator-extractor of NMZK to verify whether $x^{j\to i}_{\mathsf{NMZK}}$ is a true statement[10].

$-\mathcal{H}_3$ is equal to $\mathcal{H}_2$ with the difference that each extracted value $y^{j\to i}$, such that $Y^{j\to i}_b = f(y^{j\to i})$ for $b \in \{0,1\}$, is committed via the special non-malleable commitment scheme for all $j \in \{1,\dots,n\} \setminus \{i\}$. That is, the look-ahead rewinding threads are used to extract the trapdoor $y^{j\to i}$ using the PoK extractor of Π_{OR}. Then $y^{j\to i}$ is committed in $(\mathsf{nm}^{i\to j}_{1,1}, \mathsf{nm}^{j\to i}_{1,2}, \mathsf{nm}^{i\to j}_{1,3})$. We note that the in $\mathcal{H}_3$ the value committed inside the special non-malleable commitment can be changed when the look-ahead thread are over due to the delayed-input property of NMCOM. $\mathcal{H}_3$ and $\mathcal{H}_2$ are indistinguishable due properties of hiding and last-message pseudorandomness enjoyed NMCOM (the property last-message pseudorandomness is necessary to avoid rewinding issues during the reduction, more details will follow). In order to prove that $\text{Prob}\left[\,\mathsf{NOTRAP}_{\mathcal{H}_3}\,\right] < \nu(\lambda)$ we can rely on the property of NMCOM to be many-one synchronous non-malleable. We observe that one-one non-malleability is sufficient since, if there is one session in which the adversary commits to the trapdoor, then we can isolate that session and construct the reduction to the non-malleable commitment (a similar technique is used in [COSV16]). Note that in the reduction to the non-malleability, as well as in the reduction to the hiding of NMCOM, it is required to rewind the adversary from the third to the second round due to the look-ahead rewinding threads. However, this does not represent a problem since, during the rewinds, the reduction can compute the replies to the second round of NMCOM sent by the adversary by its own due to the property of last-message pseudorandomness. Moreover, also in this case we can rely on the simulator-extractor of NMZK to ensure that $\text{Prob}\left[\,\mathsf{FALSEX}_{\mathcal{H}_3}\,\right] < \nu(\lambda)$.

$-\mathcal{H}_4$ is equal to $\mathcal{H}_3$ with the difference that the witness used in ZAP in the main thread (after the look-ahead rewinds) is now represented by $(\mathsf{dnm}^{i\to j}_1, y^{j\to i})$ such that $Y^{j\to i}_b = f(y^{j\to i})$ and $\mathsf{Rec}(\mathsf{nm}^{i\to j}_{0,1}, \mathsf{nm}^{j\to i}_{0,2}, \mathsf{nm}^{i\to j}_{0,3}, (\mathsf{dnm}^{i\to j}_1, y^{j\to i})) = 1$ for all $j \in \{1,\dots,n\} - \{i\}$ (we recall that $y^{j\to i}_b$ is the value extracted by running the PoK extractor for Π_{OR}). Due to the WI property of ZAP we have that the two hybrid experiments are indistinguishable. In more details, during the look-ahead threads the reduction computes the ZAP proof using the same witness used in $\mathcal{H}_3$ (which corresponds to a valid defense for $\mathsf{MPC}^{\mathsf{DSM}}$ committed via the other special non-malleable commitment). When the extraction via look-ahead rewinds is over, the reduction acts as a proxy for the messages of ZAP between the ZAP challenger and $\mathcal{A}$. When the interaction with the challenger is over the reduction rewinds $\mathcal{A}$ again to check that he is still not committing to the trapdoor. We note that these rewinds do not affect the reduction due to the resettability of ZAP. The argument given above are also sufficient to claim that $\text{Prob}\left[\,\mathsf{NOTRAP}_{\mathcal{H}_4}\,\right] < \nu(\lambda)$. As for the previous hybrid experiment, we rely on the simulator-extractor of NMZK to ensure that $\text{Prob}\left[\,\mathsf{FALSEX}_{\mathcal{H}_4}\,\right] < \nu(\lambda)$.

$-\mathcal{H}_5$ is equal to $\mathcal{H}_4$ with the difference that the $j-$th flow, for $j \in \{1,\dots,n\} - \{i\}$, of NMCOM $(\mathsf{nm}^{i\to j}_{0,1}, \mathsf{nm}^{j\to i}_{0,2}, \mathsf{nm}^{i\to j}_{0,3})$ does not contain a valid defense for $\mathsf{MPC}^{\mathsf{DSM}}$ anymore in the main-thread. The indistinguishability between $\mathcal{H}_4$ and $\mathcal{H}_3$ comes from the hiding of NMCOM. We can prove that $\text{Prob}\left[\,\mathsf{NOTRAP}_{\mathcal{H}_5}\,\right] < \nu(\lambda)$ for the same arguments used above. Indeed if this does not hold, then a reduction to the security of the many-one non-malleability of NMCOM can be made. Note that in this case many-one non-malleability is sufficient since we have proved that in $\text{Prob}\left[\,\mathsf{NOTRAP}_{\mathcal{H}_4}\,\right] <$

[10] We note that the simulator of NMZK of [COSV17], in order to extract the witness for the theorem proved by the adversary just needs to rewind from the fourth to the third round, which does not affect the reduction to the PoK property of Π_{OR}.

$\nu(\lambda)$. Therefore, if there is just one session in which the adversary commits correctly to the trapdoor then a reduction can be made. We also observe that due to the soundness of ZAP, and because $\text{Prob}[\ \mathsf{NOTRAP}_{\mathcal{H}_4}\ ] < \nu(\lambda)$, then the adversary is still committing to a valid defense for $\mathsf{MPC}^{\mathsf{DSM}}$ in $(\mathsf{nm}_{0,1}^{j\to i}, \mathsf{nm}_{0,2}^{i\to j}, \mathsf{nm}_{0,3}^{j\to i})$ for all $j \in \{1,\ldots,n\} - \{i\}$ in the main-thread. As for the previous hybrid, we rely on the simulator-extractor of NMZK to ensure that $\text{Prob}[\ \mathsf{FALSEX}_{\mathcal{H}_5}\ ] < \nu(\lambda)$.

-$\mathcal{H}_6$ is equal to $\mathcal{H}_5$ with the difference that the value committed in $\mathtt{com}$ is a random value. The indistinguishability between the hybrid experiments and the fact that $\text{Prob}[\ \mathsf{NOTRAP}_{\mathcal{H}_5}\ ] < \nu(\lambda)$ come from the hiding of the commitment scheme $(\mathsf{Com}, \mathsf{Dec})$. We observe that, since $\text{Prob}[\ \mathsf{NOTRAP}_{\mathcal{H}_6}\ ] < \nu(\lambda)$, then $(\mathsf{nm}_{0,1}^{j\to i}, \mathsf{nm}_{0,2}^{i\to j}, \mathsf{nm}_{0,3}^{j\to i})$ contains a valid a defense for $\mathsf{MPC}^{\mathsf{DSM}}$ for all $j \in \{1,\ldots,n\} - \{i\}$. Therefore, due to the honest-extractability of NMCOM and the soundness of ZAP, the defense for $\mathsf{MPC}^{\mathsf{DSM}}$ can be extracted by rewinding the special non-malleable commitments that are sent from the adversary in the main-thread. We can also claim that $\text{Prob}[\ \mathsf{FALSEX}_{\mathcal{H}_6}\ ] < \nu(\lambda)$ because of the same arguments showed above.

$-\mathcal{H}_7$ is equal to $\mathcal{H}_6$ with the difference that during the look-ahead rewinding threads $\mathsf{MPC}^{\mathsf{DSM}}$ is run using a random input (like $\mathcal{H}_6$) and in the main-thread the simulator of $\mathsf{MPC}^{\mathsf{DSM}}$ is run. We note that the simulator of $\mathsf{MPC}^{\mathsf{DSM}}$ can be feed with a valid defense in the main-thread since this defense can be extracted from the non-malleable commitments received from the adversarial parties using the reusable decommitment information obtained during the look-ahead thread.

The indistinguishability between the two hybrid experiments comes from the security of $\mathsf{MPC}^{\mathsf{DSM}}$. More precisely, This part of the proof uses the same arguments proposed in Sec. 4. Indeed, in our case B is represented by $\mathsf{MPC}^{\mathsf{DSM}}$ and Ext is represented by: 1) the extractor that extracts from $(\mathsf{nm}_{0,1}^{j\to i}, \mathsf{nm}_{0,2}^{i\to j}, \mathsf{nm}_{0,3}^{j\to i})$ a defense for $\mathsf{MPC}^{\mathsf{DSM}}$ and 2) the extractor for Π_{OR} that extracts the inverses of the OWPs images sent from $\mathcal{A}$. The proof ends with the observation that this hybrid experiment corresponds to the simulator $\mathsf{Sim}_{\mathsf{MPC}}$ showed in Fig. 4. □

From Theorems 1, 2 and 3 we obtain the following corollary.

Corollary 2. *If sub-exponentially secure one-way trapdoor permutations exist, then Π^{MPC} is secure 4-round maliciously secure MPC protocol.*

Moreover, from Theorem 1, [HV16, Theorem 6.2] and Theorem 3 we can also claim the following.

Corollary 3. *If claw-free trapdoor permutations exist, then Π^{MPC} is secure 4-round maliciously secure MPC protocol.*

We assume that all but one the parties are corrupted. Let P_i be the only honest party, then $\mathsf{Sim}_{\mathsf{MPC}}$ runs internally the corrupted parties $\mathcal{P} \setminus \{P_i\}$ controlled by the adversary $\mathcal{A}$ and acts as follows.

From the 1-th to the 3-rd round.

1. Pick $x' \leftarrow \{0,1\}^\lambda$, compute $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(1^\lambda, x')$ and send $\mathtt{com}$ to all the parties.
2. Run $\mathsf{Sim}_{\mathsf{ZK}}$ on input 1^λ.
3. Pick $x, \rho \leftarrow \{0,1\}^\lambda$ and run $\mathsf{MPC}^{\mathsf{DSM}}$ in the second and the third round.
4. Commit to (x, ρ) to all the corrupted parties using NMCOM. That is, in the end of the three rounds each malicious party P_j obtains a transcript $(\mathsf{nm}_{0,1}^{i\to j}, \mathsf{nm}_{0,2}^{j\to i}, \mathsf{nm}_{0,3}^{i\to j})$ for NMCOM that represents the commitment to (x, ρ).
5. Pick $r_j \leftarrow \{0,1\}^\lambda$ and commits to it to the corrupted P_j using NMCOM for all $j \in \{1, \ldots, n\} \setminus \{i\}$. That is, in the end of the three rounds each malicious party P_j obtains a transcript $(\mathsf{nm}_{1,1}^{i\to j}, \mathsf{nm}_{1,2}^{j\to i}, \mathsf{nm}_{1,3}^{i\to j})$ for NMCOM that represents the commitment to r_j.
6. Engage a ZAP proof with each P_j in the second and the third round using as a witness $(\mathsf{dnm}_0^{i\to j}\rho, x)$.
7. Act as the receiver of NMCOM for all the non-malleable commitments received from the adversary.
8. Act as the ZAP verifier to check if all the proofs received in the third round from the adversarial parties are accepting. If there is at least one non-accepting proof then abort.

Look-ahead rewinding threads.

1. Run the extractor $\mathtt{ExtNM}$ to obtain the reusable decommitment information d_j of $(\mathsf{nm}_{0,1}^{j\to i}, \mathsf{nm}_{0,2}^{i\to j}, \mathsf{nm}_{0,3}^{j\to i})$ for all $j \in \{1, \ldots, n\} \setminus \{i\}$.
2. Run the PoK extractor of Π_{OR} thus obtaining $y^{j\to i}$ such that $f(y^{j\to i}) = Y_{b^{j\to i}}^{j\to i}$ with $b^{j\to i} \in \{0,1\}$ for all $j \in \{1, \ldots, n\} \setminus \{i\}$.

Main-thread execution. Rewind $\mathcal{A}$ up to the second round and do the following.

1. Continue the execution of $\mathsf{Sim}_{\mathsf{ZK}}$.
2. Change the committed message of the first flaw of NMCOM from (x', ρ') to a random value $r' \leftarrow \{0,1\}^\lambda$.[a]
3. Change the message committed to P_j via the second flaw of NMCOM to $y^{j\to i}$ for all $j \in \{1, \ldots, n\} \setminus \{i\}$[b].
4. Compute the j-th ZAP proof using as a witness $(\mathsf{dnm}_1^{i\to j}, y^{j\to i})$ for all $j \in \{1, \ldots, n\} \setminus \{i\}$
5. Upon receiving the third round from the adversary, let $(\mathsf{nm}_{0,1}^{j\to i}, \tilde{\mathsf{nm}}_{0,2}^{i\to j}, \tilde{\mathsf{nm}}_{0,3}^{j\to i})$ be the transcript for NMCOM obtained by interacting the the adversarial party P_j. Extract the committed value (x_j, ρ_j) using the reusable decommitment information d_j obtained during the look-ahead threads and check that (x_j, ρ_j) is a valid defense. If it is not then abort, otherwise continue with the following steps.
6. Run the simulator $\mathsf{Sim}_{\mathsf{DSM}}$ for the delayed-semi-malicious MPC protocol using as input the defence extracted from the corrupted parties (x_j, ρ_j) for all $j \in \{1, \ldots, n\} \setminus \{i\}$[c].

Output 4. Output what the $\mathcal{A}$ outputs.

[a] We recall that NMCOM is delayed-input and so the committed message can be decided in the third round of the commitment phase.

[b] In the case that $\mathcal{A}$ aborts, $\mathsf{Sim}_{\mathsf{MPC}}$ rewinds the adversary and sends a new (pseudorandom) third round of NMCOM to $\mathcal{A}$.

[c] Note that $\mathsf{Sim}_{\mathsf{DSM}}$ need the defence only to compute the last round.

Figure 4: The simulator $\mathsf{Sim}_{\mathsf{MPC}}$.

6 Acknowledgments

The second author is supported in part by NSF-BSF grant 1619348, DARPA SafeWare subcontract to Galois Inc., DARPA SPAWAR contract N66001-15-1C-4065, US-Israel BSF grant 2012366, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. The views expressed are those of the authors and do not reflect position of the Department of Defense or the U.S. Government. This research was also partially supported by H2020 project PRIVILEDGE #780477.

References

[ACJ17] Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A new approach to round-optimal secure multiparty computation. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 468–499, 2017.

[BGJ$^+$18] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 459–487. Springer, 2018.

[BHP17] Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 645–677. Springer, 2017.

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 500–532. Springer, 2018.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *In Proceedings of the International Congress of Mathematicians*, page 444âĂŞ451, 1986.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[COSV16] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Concurrent non-malleable commitments (and more) in 3 rounds. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 270–299. Springer, 2016. Full version http://eprint.iacr.org/2016/566.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Delayed-input non-malleable zero knowledge and multi-party coin tossing in four rounds. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 711–742. Springer, 2017. Full version http://eprint.iacr.org/2017/931.

[CPS+16] Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Online/offline OR composition of sigma protocols. In Marc Fischlin and Jean-Sébastien Coron, editors, *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*, volume 9666 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2016. Full version http://eprint.iacr.org/2016/175.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 283–293, 2000.

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 51–60, 2012.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In Marc Fischlin and Jean-Sébastien Coron, editors, *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*, volume 9666 of *Lecture Notes in Computer Science*, pages 448–476. Springer, 2016.

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

[GMY06] Juan A. Garay, Philip MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. *Journal of Cryptology*, 19(2):169–209, 2006.

[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.

[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. Full version: Cryptology ePrint Archive, Report 2015/1178.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 468–499. Springer, 2018.

[HHPV18] Shai Halevi, Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Round-optimal secure multi-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 488–520. Springer, 2018.

[HIK$^+$11] Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011.

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. What security can we achieve within 4 rounds? In Vassilis Zikas and Roberto De Prisco, editors, *Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings*, volume 9841 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2016. Full version: https://eprint.iacr.org/2015/797.

[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 Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 578–595. Springer, 2003.

[Lin10] Yehuda Lindell. Foundations of cryptography 89-856. http://u.cs.biu.ac.il/~lindell/89-856/complete-89-856.pdf, 2010.

[LP11] Huijia Lin and Rafael Pass. Constant-round non-malleable commitments from any one-way function. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 705–714. ACM, 2011.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, volume 4948 of *Lecture Notes in Computer Science*, pages 571–588. Springer, 2008.

[LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing,STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 179–188, 2009.

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

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 403–418, 2009.

[SCO$^+$01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. 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 566–598. Springer, 2001.

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

A Standard Definitions

Definition 14 (Computational indistinguishability). *Let $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ be ensembles, where X_λ's and Y_λ's are probability distribution over $\{0,1\}^l$, for same $l = \mathsf{poly}(\lambda)$. We say that $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are* computationally indistinguishable, *denoted $X \approx Y$, if for every* PPT *distinguisher $\mathcal{D}$ there exists a negligible function ν such that for sufficiently large $\lambda \in \mathbb{N}$,*

$$\left|\text{Prob}\left[\ t \leftarrow X_\lambda : \mathcal{D}(1^\lambda, t) = 1\ \right] - \text{Prob}\left[\ t \leftarrow Y_\lambda : \mathcal{D}(1^\lambda, t) = 1\ \right]\right| < \nu(\lambda).$$

We note that in the usual case where $|X_\lambda| = \Omega(\lambda)$ and λ can be derived from a sample of X_λ, it is possible to omit the auxiliary input 1^λ. In this paper we also use the definition of *Statistical Indistinguishability.* This definition is the same as Definition 14 with the only difference that the distinguisher $\mathcal{D}$ is unbounded. In this case use $X \equiv_s Y$ to denote that two ensembles are statistically indistinguishable.

Definition 15 (One-way function (OWF)). *A function $f : \{0,1\}^\star \to \{0,1\}^\star$ is called one way if the following two conditions hold:*

- *there exists a deterministic polynomial-time algorithm that on input y in the domain of f outputs $f(y)$;*
- *for every* PPT *algorithm $\mathcal{A}$ there exists a negligible function ν, such that for every auxiliary input $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$:*

$$\text{Prob}\left[\ y \leftarrow \{0,1\}^\star : \mathcal{A}(f(y), z) \in f^{-1}(f(y))\ \right] < \nu(\lambda).$$

We say that a OWF f is a 1-to-1 OWF *if $f(x) \neq f(y)\ \forall (x,y) \in \{0,1\}^\star$.*
We say, also, that a OWF f is a one-way permutation (OWP) *if f is a permutation.*

Definition 16 (Proof of Knowledge [LP11]). *A protocol $\Pi = (\mathcal{P}, \mathcal{V})$ that enjoys completeness is a* proof of knowledge *(PoK) for the relation $\mathsf{Rel_L}$ if there exists a probabilistic expected polynomial-time machine* E, *called the extractor, such that for every algorithm $\mathcal{P}^\star$, there exists a negligible function ν, every statement $x \in \{0,1\}^\lambda$, every randomness $r \in \{0,1\}^\star$ and every auxiliary input $z \in \{0,1\}^\star$,*

$$\text{Prob}\left[\ \langle \mathcal{P}^\star_r(z), \mathcal{V}\rangle(x) = 1\ \right] \leq \text{Prob}\left[\ w \leftarrow \mathsf{E}^{\mathcal{P}^\star_r(z)}(x) : (x,w) \in \mathsf{Rel_L}\ \right] + \nu(\lambda).$$

We also say that an argument system Π is a argument of knowledge *(AoK) if the above condition holds w.r.t. any* PPT *$\mathcal{P}^\star$.*

In our security proofs we make use of the following observation. An interactive protocol Π that enjoys the property of completeness and PoK (AoK) is a proof (an argument) system. Indeed suppose by contradiction that is not. By the definition of PoK (AoK) it is possible to extract the witness for every theorem $x \in \{0,1\}^\lambda$ proved by $\mathcal{P}^\star_r$ with probability greater than $\text{Prob}\left[\ \langle \mathcal{P}^\star_r(z), \mathcal{V}\rangle(x) = 1\ \right]$; contradiction.

In this paper we also consider the *adaptive-input* PoK/AoK property for all the protocols that enjoy delayed-input completeness. Adaptive-input PoK/AoK ensures that the PoK/AoK property still holds when a malicious prover can choose the statement adaptively at the last round.

A *3-round protocol* $\Pi = (\mathcal{P}, \mathcal{V})$ for a relation $\mathsf{Rel_L}$ is an interactive protocol played between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ on common input x and private input w of $\mathcal{P}$ s.t. $(x,w) \in \mathsf{Rel_L}$. In a

3-round protocol the first message a and the third message z are sent by $\mathcal{P}$ and the second messages c is played by $\mathcal{V}$. At the end of the protocol $\mathcal{V}$ decides to accept or reject based on the data that he has seen, i.e. $x, \mathsf{a}, \mathsf{c}, \mathsf{z}$.

We usually denote the message c sent by $\mathcal{V}$ as a *challenge*, and as *challenge length* the number of bit of c.

Definition 17 (Σ-Protocol). *A 3-round public-coin protocol $\Pi = (\mathcal{P}, \mathcal{V})$ for a relation Rel_L is a Σ-Protocol if the following properties hold:*

- *Completeness: if $(\mathcal{P}, \mathcal{V})$ follow the protocol on input x and private input w to $\mathcal{P}$ s.t. $(x, w) \in \mathsf{Rel}_\mathsf{L}$, $\mathcal{V}$ always accepts.*

- *Special soundness: if there exists a polynomial time algorithm such that, for any pair of accepting transcripts on input x, $(\mathsf{a}, \mathsf{c}_1, \mathsf{z}_1), (\mathsf{a}, \mathsf{c}_2, \mathsf{z}_2)$ where $\mathsf{c}_1 \neq \mathsf{c}_2$, outputs witness w such that $(x, w) \in \mathsf{Rel}_\mathsf{L}$.*

- *Special Honest Verifier Zero-knowledge (Special HVZK): there exists a* PPT *simulator algorithm* Sim *that for any $x \in L$, security parameter λ and any challenge c works as follow: $(\mathsf{a}, \mathsf{z}) \leftarrow \mathsf{Sim}(1^\lambda, x, \mathsf{c})$. Furthermore, the distribution of the output of* Sim *is computationally indistinguishable from the distribution of a transcript obtained when $\mathcal{V}$ sends c as challenge and $\mathcal{P}$ runs on common input x and any w such that $(x, w) \in \mathsf{Rel}_\mathsf{L}$*[11].

Definition 18. *A delayed-input 3-round protocol $\Pi = (\mathcal{P}, \mathcal{V})$ for relation Rel_L enjoys* adaptive-input special soundness *if there exists a polynomial time algorithm such that, for any pair of accepting transcripts $(\mathsf{a}, \mathsf{c}_1, \mathsf{z}_1)$ for input x_1 and $(\mathsf{a}, \mathsf{c}_2, \mathsf{z}_2)$ for input x_2 with $\mathsf{c}_1 \neq \mathsf{c}_2$, outputs witnesses w_1 and w_2 such that $(x_1, w_1) \in \mathsf{Rel}_\mathsf{L}$ and $(x_2, w_2) \in \mathsf{Rel}_\mathsf{L}$.*

A.1 Commitment Schemes

Definition 19 (Commitment Scheme). *Given a security parameter 1^λ, a commitment scheme $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ is a two-phase protocol between two* PPT *interactive algorithms, a sender* Sen *and a receiver* Rec. *In the commitment phase* Sen *on input a message m interacts with* Rec *to produce a commitment* $\mathtt{com}$, *and the private output* $\mathtt{d}$ *of* Sen.

In the decommitment phase, Sen *sends to* Rec *a decommitment information $(m, \mathtt{d})$ such that* Rec *accepts m as the decommitment of* $\mathtt{com}$.

Formally, we say that $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ is a perfectly binding commitment scheme if the following properties hold:

Correctness:

- *Commitment phase. Let* $\mathtt{com}$ *be the commitment of the message m given as output of an execution of $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ where* Sen *runs on input a message m. Let* $\mathtt{d}$ *be the private output of* Sen *in this phase.*
- *Decommitment phase*[12]. Rec *on input m and* $\mathtt{d}$ *accepts m as decommitment of* $\mathtt{com}$.

Statistical (resp. Computational) Hiding([Lin10]): *for any adversary (resp.* PPT *adversary) $\mathcal{A}$ and a randomly chosen bit $b \in \{0, 1\}$, consider the following hiding experiment* $\mathsf{ExpHiding}^b_{\mathcal{A},\mathsf{CS}}(\lambda)$:

[11] Note that we require that the two transcripts are computationally indistinguishable as in [GMY06], instead of following [CDS94] that requires the perfect indistinguishability between the two transcripts.

[12] In this paper we consider only non-interactive commitment and decommitment phase.

- *Upon input 1^λ, the adversary $\mathcal{A}$ outputs a pair of messages m_0, m_1 that are of the same length.*
- Sen *on input the message m_b interacts with $\mathcal{A}$ to produce a commitment of m_b.*
- *$\mathcal{A}$ outputs a bit b' and this is the output of the experiment.*

For any adversary (resp. PPT *adversary) $\mathcal{A}$, there exist a negligible function ν, s.t.:*

$$\left|\text{Prob}\left[\ \mathsf{ExpHiding}^0_{\mathcal{A},\mathsf{CS}}(\lambda) = 1\ \right] - \text{Prob}\left[\ \mathsf{ExpHiding}^1_{\mathcal{A},\mathsf{CS}}(\lambda) = 1\ \right]\right| < \nu(\lambda).$$

Statistical (resp. Computational) Binding: *for every commitment* $\mathtt{com}$ *generated during the commitment phase by a possibly malicious unbounded (resp. malicious* PPT*) sender* $\mathsf{Sen}^\star$ *there exists a negligible function ν such that* $\mathsf{Sen}^\star$*, with probability at most $\nu(\lambda)$, outputs two decommitments (m_0, d_0) and (m_1, d_1), with $m_0 \neq m_1$, such that* Rec *accepts both decommitments.*

We also say that a commitment scheme is perfectly binding *iff $\nu(\lambda) = 0$.*

When a commitment scheme $(\mathsf{Com}, \mathsf{Dec})$ *is non-interactive, to not overburden the notation, we use the following notation.*

– *Commitment phase.* $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(m)$ *denotes that* $\mathtt{com}$ *is the commitment of the message m and* $\mathtt{dec}$ *represents the corresponding decommitment information.*

– *Decommitment phase.* $\mathsf{Dec}(\mathtt{com}, \mathtt{dec}, m) = 1$*.*

A.2 A special WIPoK Π_{OR}.

In order to construct our MPC protocol we rely the *special WIPoK* Π_{OR} proposed in [COSV17]. This WIPoK has the property to be nicely composable with other protocols in parallel. In a nutshell, Π_{OR} takes two instantiations of the three-move Special HVZK PoK (like as in Blum's protocol [Blu86]) and composes them via the OR composition proposed in [CDS94] thus obtaining a WIPoK. Using this WIPoK a reduction can be successfully completed even when there are rewinds due to another protocols played in parallel. Since in our protocol/proof we make non-black-box use of this primitive, we provide the description of Π_{OR} verbatim from [COSV17].

Π_{OR} uses the compiler proposed in [CDS94] to combine two three-move Special HVZK PoKs Σ_0, Σ_1 into a WIPoK for the $\mathcal{NP}$-language L_0 OR L_1. Let (x_0, x_1) be the compound statement to be proved, with $x_0 \in L_0$ and $x_1 \in L_1$, and let w_b be the witness for x_b. The compiler proposed in [CDS94] executes Σ_0 and Σ_1 (respectively for L_0 and L_1) in parallel, but after receiving the challenge c from the verifier, the prover can use as challenges for Σ_0 and Σ_1 every pair (c_0, c_1) s.t. $c_0 \oplus c_1 = c$. Therefore the prover could choose in advance one of the challenge to be used (e.g., c_{1-b}), and compute the other one by setting $c_b = c \oplus c_{1-b}$. In this way the transcript for Σ_{1-b} can be computed using the Special HVZK simulator while the transcript for Σ_b is computed using the witness w_b. Thus the prover has the "freedom" of picking one out of two challenges before seeing c, but still being able to complete the executions of both Σ_0 and Σ_1 for every c. This "freedom" is sufficient to switch from the use of w_0 to the use of w_1 (in order to prove WI) even when it is required to answer to additional (and different) challenges $c^1, \ldots, c^{\mathsf{poly}(\lambda)}$ (i.e., when some rewinds occur). Indeed it is possible to switch the witness used (from w_0 to w_1) in two steps relying first on the Special HVZK of Σ_1, and then on the Special HVZK of Σ_0. More precisely, we consider the hybrid experiment H^{w_0} as the experiment where in Π_{OR} the witness w_0 is used (analogously we define H^{w_1}). We now consider H^{w_0,w_1} that differs from H^{w_0} because both the witnesses w_0 and w_1

are used. We prove that H^{w_0} and H^{w_0,w_1} are indistinguishable due to the Special HVZK of Σ_1 even though Π_{OR} is rewound polynomially many times. The reduction works as follows. A challenge c_1 is chosen before the protocol Π_{OR} starts and the Special HVZK challenger is invoked thus obtaining (a_1, z_1). The transcript for Σ_0 is computed by the reduction using the witness w_0 in order to answer to the challenge $c_0^i = c^i \oplus c_1$ for $i = 1, \dots, \mathsf{poly}(\lambda)$. We recall the we are in a setting where Π_{OR} could be rewound, and therefore the reduction needs to answer to multiple challenges. We observe that the reduction to the Special HVZK is not disturbed by these rewinds because c_1 can be kept fixed. The same arguments can be used to prove that H^{w_0,w_1} is computationally indistinguishable from H^{w_1}.

B The synchronous one-one non-malleable commitment scheme of [GPR16]

Let $(\mathsf{Com}, \mathsf{Dec})$ be a non-interactive statistically binding commitment scheme, and (E, D) be an encoding algorithm that splits the input into two codewords L and R[13]. The scheme $\mathsf{NMCOM} = (\mathsf{Sen}, \mathsf{Rec})$ proposed in [GPR16] can be described as follows.

Commitment phase. Let m be the message to be committed.

$\mathsf{Sen} \rightarrow \mathsf{Rec}$: Sen chooses $(L, R) \leftarrow \mathsf{E}(m)$ where L is viewed as a field element in $\mathbb{Z}_q$; Sen also draws $r \leftarrow \mathbb{Z}_q$ at random, compute $\mathtt{com}, \mathtt{dec} \leftarrow \mathsf{Com}(L||r)$ and send $\mathtt{com}$ to Rec.

$\mathsf{Rec} \rightarrow \mathsf{Sen}$: Rec chooses a random $\alpha \leftarrow \mathbb{Z}_q^\star$ and sends it to Sen.

$\mathsf{Sen} \rightarrow \mathsf{Rec}$: Sen sends $a = r\alpha + L$ and R to Rec.

Decommitment phase To decommit, Sen sends $\mathtt{dec}$ to Rec.

Intuitively, Sen commits to a polynomial-based 2-out-of-2 secret sharing of L in the first round, and in the third round sends R along with one share.

Honest-extractability and reusable decommitment information. As discussed in Sec 2.3 this commitment scheme is honest-extractable. Indeed, once that a complete transcript has been received, it is possible to rewind Sen from the third to the second round using different second messages thus obtaining L. More precisely, let $\tau = (\mathtt{com}, \alpha, (a, R))$ be an honestly generated transcript. Once that another honest transcript is obtained L can be extracted (via interpolation), and the committed message can be retrieved by running D on input (L, R). We note that the extracted information can be reused to retrieve the message committed in another transcript of NMCOM that shares the same first round. For example, one can infer that the message committed in the (honestly generated) transcript $\tilde{\tau} = (\mathtt{com}, \tilde{\alpha}, (\tilde{a}, \tilde{R}))$ is $\mathsf{D}(L, \tilde{R}) = \tilde{m}$.

[13] L and R are codewords of a non-malleable code. In this discussion we can consider (E, D) just as an encoding-decoding algorithm.