Four-Round Black-Box Non-malleable Schemes from One-Way Permutations

Four-Round Black-Box Non-Malleable Commitments from One-Way Permutations

Michele Ciampi[1], Emmanuela Orsini[2], and Luisa Siniscalchi[3]

[1] The University of Edinburgh, Edinburgh, UK
[2] imec-COSIC, KU Leuven, Leuven, Belgium.
[3] Dept. Computer Science, Aarhus University, Aarhus, Denmark.
michele.ciampi@ed.ac.uk, emmanuela.orsini@kuleuven.be, lsiniscalchi@cs.au.dk

Abstract. We construct the first four-round non-malleable commitment scheme based solely on the black-box use of one-to-one one-way functions. Prior to our work, all non-malleable commitment schemes based on black-box use of polynomial-time cryptographic primitives require more than 16 rounds of interaction.
A key tool for our construction is a proof system that satisfies a new definition of security that we call *non-malleable zero-knowledge with respect to commitments*. In a nutshell, such a proof system can be safely run in parallel with a (potentially interactive) commitment scheme. We provide an instantiation of this tool using the MPC-in-the-Head approach in combination with BMR.

1 Introduction

Starting from the pioneering work of Dolev et al. [DDN91], a long line of works has focused on constructing new non-malleable commitment schemes with improved characteristics, both in terms of efficiency and assumptions. Given the strong connection of non-malleable commitments with secure multi-party computation [Yao82, BMR90], improvements in the area of non-malleable commitments have a big impact on the multi-party computation (MPC) landscape. In particular, recent developments on the round complexity of non-malleable commitments led to the first round-optimal MPC protocols in the plain model [COSV17b, HHPV18, BGJ+18, CCG+20].

The round complexity of commitment schemes based on polynomial-time hardness assumptions in the stand-alone setting is nowadays well understood. Non-interactive commitments can be constructed assuming the existence of 1-to-1 one-way functions (OWFs) [GL89] and 2-round commitments can be constructed assuming the existence of OWFs only. Moreover, non-interactive commitments do not exist if one relies on the black-box use of OWFs only [MP12]. Recently many progress have been made also for the case of *non-malleable* (NM) commitments[4]. Indeed, the long sequence of very exciting positive results [Bar02, PR03, PR05b, PR05a, PR08b, PR08a, LPV08, PW10, Wee10, LP11, LP15, Goy11, GLOV12, GRRV14a, COSV17a, COSV16] led to the work of Khurana [Khu17] in which the authors showed how to obtain a 3-round (which is optimal for the case of polynomial-time assumptions [Pas13]) non-malleable commitment scheme based on specific number-theoretic assumptions, and to [GR19] where the authors proposed a round optimal scheme based on one-to-one OWFs.

Black-box (BB) constructions. While these recent results show round-optimal constructions, they make non-black-box use of cryptography. Constant round BB schemes are known [Wee10, LP11, Goy11, GLOV12], but their round complexity is far to optimal. More specifically, Goyal et al.

[4] In this paper we will consider only NM commitments w.r.t. commitments. For the case of NM w.r.t. decommitments see [PR05b, PR08b, OPV09, CVZ10, DMRV13, GKS16].

[GLOV12] give a black-box NM commitment protocol only based on the existence of one-way functions, but this construction requires more than 16 rounds. In another work, Goyal et al. [GRRV14a] mention that combining their protocol with ideas from [GLOV12] could lead to a 6-round protocol, but no explicit construction or proof intuition was given. Therefore the following question remained open.

> *Does it exist a non-malleable commitment scheme that makes black-box use of standard polynomial-time cryptographic primitives where the commitment phase consists of less than 16 rounds?*

In this work, we provide a positive answer, by proposing a 4-round non-malleable commitment scheme that only makes black-box use of one-to-one one-way functions. Whether it is possible to achieve the same result in three rounds remains a fascinating open question.

1.1 Our Contributions

The state-of-the-art in constructing non-malleable commitments based on minimal assumptions shows a significant gap in the round complexity of black-box and non-black-box protocols. In this work, we almost close this gap by describing the first 4-round non-malleable commitment that makes black-box use of the underlying primitives and is based on the almost minimal assumption of injective one-way functions.[5] In particular, we prove the following theorem.

Theorem (Informal). *Assuming one-to-one OWFs, there exists a 4-round non-malleable commitment scheme that makes black-box use of the OWFs.*

Our 4-round non-malleable commitment crucially relies on a novel 3-round public-coin proof system that is zero-knowledge against honest verifiers (HVZK), and such that the statement to be proven can be specified in the last round (*delayed-input property*). In particular, our protocol enjoys *adaptive-soundness* and *adaptive-HVZK* [HV16, CPS+16, CPV20]. These properties guarantee that HVZK and soundness hold even against an adversary that decides the statement to be proven (and the witness for the HVZK case) adaptively on the first two rounds of the protocol. A protocol that satisfies such properties and that also makes black-box use of the underlying cryptographic primitives is proposed in [HV16]. What makes our scheme different is that it also enjoys a special form of non-malleability that we call *non-malleable HVZK with respect to commitment (NMZKC)*.

In a nutshell, this notion allows us to safely compose the proof system in parallel with any type of commitment scheme. In more detail, we consider the following setting. There is a man-in-the-middle (MiM) adversary that interacts (acting as the verifier) with an honest prover of a proof system Π_{AI} (where AI stands for adaptive-input). In the right session instead, the MiM acts as the sender for a (potentially interactive) commitment scheme $\Pi_{\mathtt{com}}$, with an honest receiver. The notion of NMZKC guarantees that the distribution of the messages committed by the MiM in the right session is independent of whether the messages of Π_{AI} are generated honestly (i.e., using the witness for some NP statement x), or are computed using the simulator.

We believe that this tool and notion can be of independent interest. Indeed, NMZKC proof systems might be used in place of *rewind secure* schemes. A rewind secure proof system guarantees

[5] Our BB 4-round non-malleable commitment scheme satisfies the notion of standalone (or one-one) non-malleability. Obtaining a concurrent (or many-many) BB non-malleable commitment scheme in just 4 rounds, or less, still remains an open question.

that the zero-knowledge property holds even if an adversarial verifier is allowed to rewind the prover a bounded number of times (this can be seen as a mild form of resettability). The reason why the notion of rewind security has gained a lot of attention recently is exactly that it simplifies the composition of proof systems with other primitives. For example, it simplifies the composition of a proof system with extractable commitments. The high-level idea is that in the security proof it is possible to extract from the commitment without harming the zero-knowledge property of the proof system. Hence, it is possible to check whether the distribution of the committed messages changes depending on whether the messages of the proof system are simulated or are generated honestly. This proof technique has been exploited in many recent works [CCG+20, GR19, CRSW22]. And, more interestingly, it was used also to construct the first one-one non-malleable commitment [GRRV14b][6]. As we will discuss in the technical overview, we will replace the rewind secure proof system proposed in [GRRV14b] (that inherently makes non-black-box use of the underlying primitives) with our NMZKC proof system.

We believe that NMZKC in some scenarios can replace the use of rewind secure primitives, and this might be particularly helpful given that our protocol is completely black-box in the use of the underlying cryptographic primitives. In summary, we prove the following theorem.

Theorem (Informal). *Assuming one-to-one OWFs, then there exists a 3-round delayed-input public-coin adaptive-input proof system that also is NMZKC and it makes black-box use of the OWFs.*

2 Overview of Techniques

We first describe how to construct the main tool required for our construction, which is a commit-and-prove proof system that satisfies the definition of non-malleable HVZK with respect to commitment. Then we show how to use this tool to construct our four-round non-malleable commitment protocol.

2.1 Our NMZKC Protocol and New Commitment Schemes

We start this section by recalling how to turn an MPC protocol into a proof system for any $\mathcal{NP}$-relation Rel following the *MPC-in-the-head* approach of [IKOS07]. Let Π_{MPC} be an n-party MPC protocol that is secure against up to t semi-honest corruptions. First, the prover secret-shares the witness w using an additive secret-sharing, while f will be a verification function that outputs 1 iff w is a valid witness, i.e., $f(x, w_1, \ldots, w_n) = 1 \iff (x, w_1 \oplus \cdots \oplus w_n) \in \mathsf{Rel}$. Then, it simulates all n parties running the protocol locally and sends the verifier commitments to each parties' views. Later, the verifier randomly chooses t of the parties' commitments to be opened, and checks that the committed messages are consistent with an honest execution of the MPC protocol according to the opened views. Since only t parties are opened, the verifier learns nothing about the secret input w, while the random choice of the opened parties ensures that enough views have been computed honestly, ensuring soundness.[7]

[6] In Section 9 we propose a comparison between the approach based on rewind-secure primitives of [GRRV14b] and the one we propose in this work. In particular, we explain why and how we can rely on a simpler underlying weak-non-malleable commitment scheme compared to the one used in [GRRV14b].

[7] This sketch protocol gives a noticeable probability of cheating to the prover, typically the soundness of the protocol can be easily amplified via parallel repetition.

Unfortunately, this scheme is inherently non-delayed input since the prover needs both statement and witness to generate the views that must be committed in the first round. To overcome this limitation, we consider a specific class of two-phase MPC protocols. In particular, we require protocols with an input-independent offline phase, where the parties only produce correlated randomness that will be used to speed up the second phase. In the second phase (the online phase) the input is required and used to compute the output of the function. We denote such protocols by $\Pi_{\mathsf{MPC}} := (\Pi^{\mathtt{off}}_{\mathsf{MPC}}, \Pi^{\mathtt{on}}_{\mathsf{MPC}})$, where the two algorithms $\Pi^{\mathtt{off}}_{\mathsf{MPC}}$ and $\Pi^{\mathtt{on}}_{\mathsf{MPC}}$ denote respectively the offline and the online phase of Π_{MPC}.

Equipped with such an MPC protocol, we can modify the approach of [IKOS07] as follows. The prover only simulates $\Pi^{\mathtt{off}}_{\mathsf{MPC}}$, and commits to the individual views. Then the verifier, as described before, selects a random subset of parties to be opened. After receiving the challenge, the prover opens the requested commitments and additionally runs $\Pi^{\mathtt{on}}_{\mathsf{MPC}}$ to obtain the entire views of the parties requested by the verifier. At the end of this process, the verifier holds complete views for all the parties it requested and can check their consistency as previously described.

Intuitively, (non-adaptive input) HVZK comes again from the hiding of the commitments and the (semi-honest) security of the MPC protocol. However, it is clear that this approach fails completely against malicious provers. Indeed, they might easily generate online messages in a malicious way for all the parties the verifier did not ask to open. Note that in this case, Π_{MPC} is secure against t corrupted parties, but the adversary might generate ill-formed online messages for the remaining $n-t$. To work around this problem, we require Π_{MPC} to enjoy a stronger notion of security that we call *robustness*. In a nutshell, this notion requires that, when the offline phase of Π_{MPC} has been honestly computed, then it is always possible to check if a message received during the online phase has been honestly generated or not. In this way, robustness allows to prove soundness also w.r.t. a malicious prover that specifies the inputs in the last round (i.e. adaptive-input soundness).

The above approach guarantees that the protocol enjoys delayed-input completeness and adaptive-input soundness. However, it is not clear how to argue that the protocol is adaptive-input HVZK given that Π_{MPC} is only semi-honest secure. The reason is that we would like to rely on the security of the underlying MPC protocol thus committing to simulated views in the first round. However, to simulate these views the MPC simulator needs to know the input of the corrupted parties. We recall that such input consists of a share of the witness (which is easy to simulate) and the theorem to be proven. This is problematic since the adaptive-input HVZK simulator needs to generate the first round without knowing the theorem, hence, we cannot run the MPC simulator of the underlying protocol.

To circumvent this issue, we make use of a special type of commitment scheme, that we call *ambiguous* commitment[8]. Compared to a standard commitment scheme, they have two modes of operation: binding and equivocal. If the commitment is computed using the binding mode then the commitment is binding, otherwise, it can be equivocated to any message the sender wants.

Using ambiguous commitments, we modify our protocol as follows. The prover generates the views of Π_{MPC} as before, but it creates a 2-out-of-2 secret sharing of each of these views and commits to them using the ambiguous commitment scheme in biding mode (i.e., two commitments per view are generated). Then, the verifier challenges the prover asking to open a random subset of views as before. In addition, for each of the opened views, the verifier asks to see the randomness used to generate one of the two commitments and rejects if it notices that a commitment has not been computed using the binding procedure. The rest of the protocol proceeds as before.

[8] Such commitments are sometimes called *equivocal* or *trapdoor* commitments

The adaptive-input HVZK simulator, which we recall needs to generate the first round without knowing the theorem, works as follows. On input the challenge it can compute one commitment in equivocal mode (the one for which the simulator will not need to disclose its randomness), and one in binding mode. The binding commitments simply contain a random string. The set of commitments computed in the described way constitutes the first round.

Upon receiving the theorem, the adaptive-input HVZK simulator runs the MPC simulator of Π_{MPC}. At this point, the simulator computes the xor of the i-th view with the random string committed in the i-th binding commitment and opens the equivocal commitment to the obtained value.

The soundness still holds because, intuitively, the verifier performs a cut-and-choose to make sure that the commitments are all computed in binding mode. Clearly, an adversary has still a non-negligible probability of cheating, but by repeating the protocol we obtain a sound protocol.

Non-Malleable HVZK with respect to Commitment. So far we have only argued that our protocol, that we denote with Π_{AI}, is adaptive HVZK and adaptive sound. We also want to argue that our protocol is non-malleable HVZK with respect to commitment. We recall that in this security notion, there is a MiM adversary that on the left session acts as the adversary for the adaptive HVZK security game, and in the right session it acts as the sender for a commitment scheme. In more detail, the adversary picks a challenge and sends it to the left session (that acts as a challenger for the experiment). The challenger tosses a coin b, and if $b = 0$ then it computes the first round of Π_{AI} using the honest prover procedure, otherwise it computes it using the adaptive HVZK simulator. The adversary now picks a statement x and a witness w and sends those to the challenger. If $b = 0$, the challenger runs the honest prover of Π_{AI} on input (x, w) to compute a third-round message, if $b = 1$ instead the challenger runs the HVZK on input x (and the previous state of the simulator), thus obtaining the third message. The challenger then sends this third message to the MiM in the left session and stops.

While the MiM is acting as described in the left session, it concurrently sends a commitment in the right session. We say that Π_{AI} is non-malleable HVZK with respect to commitment, if the distribution of the messages committed on the right session by the MiM does not depend on b.

We prove that Π_{AI} is non-malleable HVZK with respect to any extractable commitment $\Pi_{\mathtt{com}}$. The idea is to use an adversary to the NMZKC property to construct an adversary for the adaptive-HVZK property. That is, we let the MiM to interact with the adaptive HVZK challenger while at the same time we run the extractor of the commitment scheme to check how the distribution of the committed messages changes. Unfortunately, this simple idea has a major flaw. The rewinds made by the extractor of the commitment might also rewind the challenger of the HVZK security game. Indeed in each rewind made by the extractor, the MiM could send a new theorem-witness pair, and ask for a new third round of Π_{AI}.

To prove that Π_{AI} can cope with such an adversarial behavior, we exploit how our HVZK simulator works. We note that once the challenge is known, then the simulator knows what commitments will be opened to honestly and what commitments will be equivocated. If an adversary during the rewinds samples new theorem-witness, we simply need to run multiple times the simulator of the underlying MPC protocol and equivocate the commitments accordingly. Hence, we can reduce the adversary that wins in the non-malleable HVZK with respect to commitment experiment to an adversary that either breaks the security of our commitment or the security of the underlying MPC protocol.

Σ-Commitment. In this work, we also consider a class of three-round public commitment schemes that we call Σ-commitment. A Σ-commitment is hiding against honest receiver (HRH), and in addition, it is extractable. To realize a Σ-commitment $\Sigma = (\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma})$, we use the approach of Goyal et al. [GLOV12], which makes use of an information-theoretic verifiable secret sharing protocol Π^{vss}. The protocol works as follows. To commit to a message w, the sender $\mathcal{S}^{\Sigma}$ runs "in its head" the sharing phase of Π^{vss}, with input a message m. Then the sender commits to the views (obtained by the execution of sharing phase of Π^{vss}) of each player separately using a statistical binding commitment scheme $\Pi^{\mathtt{com}}$. The receiver, upon receiving these commitments, samples a random set $I \subset [n]$, with $|I| \leq t$, and sends it to the sender. Finally, the sender replies by decommitting the views corresponding to the challenge I.

The property of HRH comes from the fact that, if the challenge I is known in advance, then we can commit to a random message and simulate the openings of the commitment. We can prove that a simulated transcript is indistinguishable from the transcript generated by an honest committed with input m via a simple reduction to the security of the statistically binding commitments.

Putting together Σ and Π_{AI} to realize a commit-and-prove protocol Π. We use Σ and Π_{AI} to realize a black-box commit-and-prove protocol, which will be the main building block we use to construct our non-malleable commitment scheme. Our commit-and-prove protocol Π works as follows. The prover commits λ-times to the witness w running Σ and proving, using Π_{AI}, that each committed message w satisfies some relation Rel[9]. The statement to be proven can be postponed to the last round since Π_{AI} is delayed-input complete.

To make sure that the same message is committed in all these executions, we use a technique proposed by Khurana et al. in [KOS18]. Namely, in each execution of Σ, instead of committing to w, we commit to $w||r$, for some random value r. Then, we use the protocol Π_{AI} to prove that $a = w + r\alpha$, where α is chosen as part of the challenge, and a is sent in the third round from the prover.

As argued in [KOS18], since r is global across all the executions, if $w \neq w'$ then $w + r\alpha \neq w' + r\alpha$ with overwhelming probability due to the Schwartz-Zippel lemma. Therefore, if the committed messages are different across the (multiple) executions, then the statement proven by Π_{AI} must be false, and the soundness of Π_{AI} guarantees that the verifier rejects. The adaptive-input SHVZK follows from the adaptive-input SHVZK of Π_{AI} and the HRH property of Σ.

Concrete instantiation for robust MPC. As we mentioned, one of the main tool we rely on is a robust MPC protocol. We recall that a robust MPC protocol allows the prover to initially commit only to the offline views, which are input-independent, and only in the last round to "complete the proof" with the online views. The robustness property guarantees that the commitments generated in the first round univocally specify the actual MPC evaluation so that the online steps only consist of an input-distribution phase and deterministic computations. In this way, even if the prover already knows which views are going to be opened, it cannot force the evaluation to output 1 unless $\mathsf{Rel}(x, w) = 1$, except with negligible probability.

Although robustness seems a very strong requirement, we show that a minor modification of the standard BMR protocols leads to an efficient robust MPC scheme. We recall that BMR [BMR90] is

[9] Π_{AI} works for any type of secret sharing scheme, and in our case Π_{AI} is parametrized by the reconstruction algorithm of the verifiable secret sharing Π^{vss} (i.e., the prover of Π_{AI} expects to receive n views generated using the sharing algorithm of Π^{vss}). We note that given that Π^{vss} is information-theoretic, then Π_{AI} still makes black-box use of the underlying cryptographic primitives.

a two-phase protocol consisting of an input-independent phase, also called *garbling*, and an online evaluation. In the garbling step, all parties $\mathtt{P}_1, \ldots, \mathtt{P}_n$ involved in the protocol generate a sharing of the garbled circuit according to some fixed secret sharing scheme $\langle \cdot \rangle$ with t-privacy. As in any other garbled-circuit based scheme, to garble a Boolean circuit each wire is assigned two random keys $\boldsymbol{k}_{w,0}, \boldsymbol{k}_{w,1}$ encoding, respectively, the 0-value and 1-value. The goal of the process is to generate four ciphertexts for each gate according to the gate function, such that each output-wire key is encrypted according to all combinations of input-wire keys which evaluate to that output wire key. During the online evaluation, these encrypted truth tables, are revealed to all parties so to allow local evaluation of the circuit. Intuitively, it is clear that upon collecting all the input keys, parties can start evaluating the circuit. At this point, this evaluation is completely deterministic and does not require any interaction. For this reason, assuming that the garbling phase is correctly generated and the input-keys corresponding to the input-wires of the circuit are correct, namely they correspond to the keys generated in the offline phase, the online views generated by each party correspond to a correct evaluation of the garbled circuit and cannot lead to an incorrect result. In the next sections, we will recall the basics of BMR-style protocols and explain the robustness property in more detail.

2.2 4-Round Non-Malleable Commitment Π_{nmc}

We are finally ready to describe how our non-malleable commitment scheme works. Our starting point is the 3-round public-coin commitment scheme of Goyal et al. [GRRV14a]. This commitment scheme, which we denote with Π_{wnmc}, is non-malleable against adversaries that never commit to $\perp$ (i.e., the adversary always generates well-formed commitments). In [GRRV14b] to lift the security of such a commitment and build a fully non-malleable commitment scheme, the authors run, in parallel with Π_{wnmc}, a zero-knowledge proof.

As noted in [GRRV14b, COSV17a], a standard ZK proof does not suffice since the commitment and the zero-knowledge proof might not be composed in parallel. As such, and as we have already anticipated, in [GRRV14a] the authors rely on a ZK proof that is rewind-secure. We also note that the statement to be proven by the ZK is fully-formed only in the last round (since Π_{wnmc} consists of 3 rounds.) This inherently requires the ZK protocol to be delayed-input. To the best of our knowledge, the only protocols that satisfy all these properties are that proposed in [GR19, GRRV14b], which, unfortunately, make non-black-box use of the underlying primitives. In [COSV17a], the authors propose a ZK proof that can be composed in parallel with the weak-non-malleable commitment of Goyal et al., but this approach requires non-black-box access to the commitment scheme.

The idea is to use our commit-and-prove protocol Π, and argue that it can be safely composed in parallel with Π_{wnmc} due to the property of NMZKC. In particular, in the security proof, we can switch from using the honest prover procedure of Π to the simulated one while making sure that the adversary cannot change what he is committing. Unfortunately, Π is only honest-verifier zero-knowledge, and here we need a zero-knowledge proof that is secure against any type of adversaries.

To lift the security of our protocol, we rely on the FLS-trick [FLS90] (with some modifications). More concretely, we construct a 4-round zero-knowledge protocol as follows. The verifier generates two commitments of two random strings, $\hat{s}_0$ and $\hat{s}_1$ in the first round and sends two openings in the third round. In parallel, the verifier provides a witness indistinguishable (WI) proof, Π_{comWI}, which guarantees that at least one of the two commitments is binding. In [KOS18], the authors show how to obtain this protocol in a black-box-way. The prover instead uses a 3-round public-coin WI to prove that either the commitment Π_{wnmc} is well-formed or that it committed to $\hat{s}_b$, for some

$b \in \{0,1\}$. Since the receiver discloses $\hat{s}_0, \hat{s}_1$ only in the last round, the sender has no way to commit (already in the second round), to either of these two values. As such, the (potentially corrupted) sender, can complete an accepting WI proof only by proving that the non-malleable commitment is well-formed. For more detail, we refer to the technical part of the paper.

3 Preliminaries

Here we recall some preliminaries that will be useful in the rest of the paper. Let λ denote the security parameter and $\mathsf{negl}(\lambda)$ any function which tends to zero faster that λ^{-c}, for any constant c. We write $[n]$ to denote the set $\{1, \ldots, n\}$. We use the abbreviation PPT to denote probabilistic polynomial-time.

Let $\mathcal{S}$ and $\mathcal{R}$ two interactive algorithms, we denote by $\langle \mathcal{S}(x), \mathcal{R}(y) \rangle(z)$ the distribution of $\mathcal{R}$'s output after an interaction with $\mathcal{S}$ on common input z and private inputs x and y. A *transcript* of $\langle \mathcal{S}(x), \mathcal{R}(y) \rangle(z)$ consists of all the messages exchanged during an interaction between $\mathcal{R}$ and $\mathcal{S}$.

3.1 Commitment Schemes

A commitment scheme $\Pi_{\mathtt{com}} = (\mathcal{S}, \mathcal{R})$ is a two-phase protocol between two PPT interactive algorithms, a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. In the first phase, called *commit phase*, $\mathcal{S}$ on input a message m interacts with $\mathcal{R}$. Let $\mathtt{com}$ be the transcript of this interaction. In the second phase, called *decommitment phase*, the sender $\mathcal{S}$ reveals m' and $\mathcal{R}$ accepts the value committed to be m' if and only if $\mathcal{S}$ proves that $m = m'$. Typically, a commitment scheme satisfies two main properties: informally, the *binding* property ensures that $\mathcal{S}$ cannot open the commitment in two different ways; the *hiding* property guarantees that the commit phase does not reveal any information about the message m. We refer the reader to [Gol06] for more details.

3.2 Extractable Commitments

Informally, a commitment scheme is said to be extractable if there exists a PPT extractor that can extract the committed value with guaranteed correctness of extraction. In particular, if the commitment is maliciously generated, then the extractor must output $\perp$, while if the commitment is honestly computed, then the extractor must output the correct value.

Definition 1 (Extractable commitment). *Consider any statistically binding, computationally hiding commitment scheme $\Pi_{\mathtt{comExt}} = (\mathcal{S}, \mathcal{R})$. Let $\tau = \mathsf{Trans}(\mathcal{S}(m, r_s), \mathcal{R}(r_r))$ denote a commitment transcript with committer input m, committer randomness r_s and receiver randomness r_r, and let $\mathsf{Dec}(\tau, m, r_s)$ denote the algorithm that on input a commitment transcript τ, committer message m and randomness r_s outputs 1 or 0 to denote whether or not the decommitment was accepted. Then $\Pi_{\mathtt{comExt}} = (\mathcal{S}, \mathcal{R})$ is said to be* extractable *if there exists an expected PPT oracle algorithm E (the extractor), such that for any PPT cheating committer $\mathcal{S}^\star$ the following holds. Let $\mathsf{Trans}(\mathcal{S}^\star, \mathcal{R}(r_r))$ denote a (potentially maliciously generated) transcript of the interaction between $\mathcal{S}^\star$ and $\mathcal{R}$. Then $E^{\mathcal{S}^\star}\big(\mathsf{Trans}(\mathcal{S}^\star, \mathcal{R}(r_r))\big)$, with oracle access to $\mathcal{S}^\star$, outputs m such that, over the randomness of E and of sampling $\mathsf{Trans}(\mathcal{S}^\star, \mathcal{R}(r_r))$,*

$$\Pr[(\exists \tilde{m} \neq m, \tilde{r}_s) : \mathsf{Dec}(\tau, \tilde{m}, \tilde{r}_s) = 1] = \mathsf{negl}(\lambda).$$

Definition 2 (k-Extractable Commitments). *An extractable commitment satisfying Definition 1 is said to be* k-extractable *if there exists a polynomial $p(\cdot)$ such that the extractor E with $(k-1)$ queries to $\mathcal{S}^\star$ outputs (m, r_s) such that over the randomness of E and of sampling $\mathsf{Trans}(\mathcal{S}^\star, \mathcal{R}(r_r))$:*

$$\Pr[\exists r_s : \mathsf{Dec}(\tau, m, r_s) = 1] \geq p(\lambda).$$

3.3 Non-Malleable Commitments

Here we follow the same notation of Goyal et al. [GRRV14a]. Let $\Pi = (\mathcal{S}, \mathcal{R})$ be a statistically binding commitment scheme and let λ be the security parameter. Consider a man-in-the-middle (MiM) adversary $\mathcal{A}$ that is participating in two interactions called the left and the right interaction. In the left interaction $\mathcal{A}$ is the receiver and interacts with an honest committer $\mathcal{S}$, whereas in the right interaction $\mathcal{A}$ is the committer and interacts with an honest receiver $\mathcal{R}$.

We compare between a MiM execution and a simulated execution.

In the MiM execution the adversary $\mathcal{A}$, with auxiliary information z, is simultaneously participating in a left and right session. In the left sessions, the MiM adversary $\mathcal{A}$ interacts with $\mathcal{S}$ receiving commitments to values $m_i, i \in [\mathsf{poly}(\lambda)]$, using identities tg_i of its choice. In the right session, $\mathcal{A}$ interacts with $\mathcal{R}$ attempting to commit to related values $\tilde{m}_i$ again using identities of its choice $\widetilde{\mathsf{tg}}_i$. If any of the right commitments is invalid, or undefined, its value is set to $\perp$. For any i such that $\mathsf{tg}_i = \mathsf{tg}_j$, for some j, set $\tilde{m}_i = \perp$ (i.e., any commitment where the adversary uses the same identity of the honest sender is considered invalid). Let $\mathsf{mim}_\Pi^{\mathcal{A},\boldsymbol{m}}(z)$ denote a random variable that describes the values $\tilde{m}_i$ and the view of $\mathcal{A}$, in the above experiment.

In the simulated execution, an efficient simulator Sim directly interacts with $\mathcal{R}$. Let $\mathsf{sim}_\Pi^{\mathsf{Sim}}(1^\lambda, z)$ denote the random variable describing the values $\tilde{m}_i$ committed by $\mathcal{A}$, and the output view of Sim; whenever the view contains in the right session the same identity of any of the identities of the left session, then m is set to $\perp$.

In all the paper we denote by $\tilde{\delta}$ a value associated with the right session (where the adversary $\mathcal{A}$ plays with a receiver) where δ is the corresponding value in the left session. For example, the sender commits to v in the left session while $\mathcal{A}$ commits to $\tilde{v}$ in the right session.

Definition 3 (Non-Malleable (NM) commitment scheme [GRRV14a]). *A commitment scheme is* NM with respect to commitment *if, for every* PPT *MiM adversary $\mathcal{A}$, there exists a* PPT *simulator* Sim *such that for all $\boldsymbol{m} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following ensembles are computationally indistinguishable:*

$$\{\mathsf{mim}_\Pi^{\mathcal{A},\boldsymbol{m}}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{sim}_\Pi^{S}(1^\lambda, z)\}_{z\in\{0,1\}^\star}.$$

In this work, we also consider a weaker class of MiM adversaries called *synchronizing adversaries*. A synchronizing adversary is one that sends its message for every round before obtaining the honest party's message for the next round.

3.4 Σ-Commitments

We now give the definition of Σ-commitment. This notion is reminiscent of the notion of Σ-protocols.

Definition 4. *A* Σ-commitment *$\Pi^\Sigma = ((\mathcal{S}^\Sigma, \mathcal{R}^\Sigma), \mathsf{Dec}^\Sigma)$ is a commitment scheme where: 1) The commitment phase consists of three rounds and it is public-coin, 2) The decommitment phase is non-interactive, and 3) It satisfies the following properties.*

– Correctness. *Let m be the message the sender $\mathcal{S}^{\Sigma}$ uses during the commitment phase. If both $\mathcal{S}^{\Sigma}$ and $\mathcal{R}^{\Sigma}$ follow the protocol, then the receiver always accepts the commitment as valid. Moreover, if the sender follows the protocol during the decommitment procedure Dec^{Σ} then the receiver accepts m as the committed message.*
– Honest Receiver Hiding (HRH). *For any message $m \in \{0,1\}^{\ell}$, there exists a polynomial-time simulator Sim, which on input a random c (sampled from the space of all the possible $\mathcal{R}^{\Sigma}$'s messages), outputs an accepting commitment transcript of the form (a, c, z) that is computationally indistinguishable from the transcript generated by the honest sender and receiver when the receiver uses m as its input (note that Sim needs to generate the transcript without knowing m).*
– t-Special Binding. *From any set of t accepting transcripts $\{a, c_i, z_i\}_{i\in[t]}$, with $c_i \neq c_j$ for all $i, j \in [t]$, for the commitment phase it is possible to extract the message m in polynomial-time, where m is the only possible message that the (potentially corrupted) sender can decommit to.*

3.5 Ambiguous Commitment Scheme

Here we formalise the definition of ambiguous commitments. Loosely speaking, they allow to overcome the binding property and "decommit ambiguosly" to any value. Compared to standard commitment schemes, we have two additional algorithms $\mathsf{Com}^{\mathsf{eq}}$ and Eq. The first one takes as input some randomness r and the length ℓ of messages, and outputs a "fake" commitment $\widehat{\mathtt{com}}$ that is not associated to any message; for any message $m \in \{0,1\}^{\ell}$, Eq on input the same randomness r used in $\mathsf{Com}^{\mathsf{eq}}$ generates a decommitment associated with m and $\widehat{\mathtt{com}}$.

Definition 5. *An* ambiguous commitment scheme *$\Pi_{\mathtt{com}}$ is defined by four algorithms $(\mathsf{Com}, \mathsf{Dec}, \mathsf{Com}^{\mathsf{eq}}, \mathsf{Eq})$ with the following syntax.*

– *The algorithm $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(m; R)$ takes as inputs a message $m \in \{0,1\}^{\ell}$ and randomness $R \xleftarrow{\$} \{0,1\}^{\lambda}$ and outputs a commitment $\mathtt{com}$ and an opening value $\mathtt{dec}$.*
– *The algorithm $\widehat{\mathtt{com}} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^{\ell}; r)$ takes as input a random coin $r \xleftarrow{\$} \{0,1\}^{\lambda}$ and outputs a commitment $\widehat{\mathtt{com}}$.*
– *The algorithm $\widehat{\mathtt{dec}} \leftarrow \mathsf{Eq}(\widehat{\mathtt{com}}, r, m)$ takes as inputs $\widehat{\mathtt{com}}$, the same randomness r used to generate $\widehat{\mathtt{com}}$ and any message $m \in \{0,1\}^{\ell}$, and outputs an opening value $\widehat{\mathtt{dec}}$.*
– *The algorithm $\mathsf{Dec}(\mathtt{com}, m, \mathtt{dec}) = b$ takes inputs $\mathtt{com}, m, \mathtt{dec}$ and outputs a bit $b \in \{0,1\}$. In particular, $b = 1$ if $\mathtt{dec}$ is a valid opening of the commitment, and $b = 0$ otherwise.*

Notice that we require that when the commitments are honestly generated following the procedure Com, then they satisfy the standard hiding and binding properties. Formally, the algorithms satisfy the following properties.

(Perfect) Correctness: *For every $m \in \{0,1\}^{\ell}$ and $R \in \{0,1\}^{\lambda}$,*

$$\Pr[\mathsf{Dec}(\mathtt{com}, m, \mathtt{dec}) = 1 : (\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(m; R)] = 1.$$

Note this property holds for every $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(m; R)$ and also for every $(\widehat{\mathtt{com}}, \widehat{\mathtt{dec}})$ such that $\widehat{\mathtt{com}} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^{\ell}; r)$ and $\mathtt{dec} \leftarrow \mathsf{Eq}(\widehat{\mathtt{com}}, r, m)$.

Binding: *For every* honestly generated commitment $\mathtt{com}$ *using procedure* Com *and randomness* $R \in \{0,1\}^\lambda$ *and every* PPT *adversary* $\mathcal{A}$ *with auxiliary information* z*, there exists a negligible function* negl *such that:*

$$\Pr\left[\begin{array}{c}\mathsf{Dec}(\mathtt{com}, m_1, \mathtt{dec}_1) = 1 \text{ and } \mathsf{Dec}(\mathtt{com}, m_2, \mathtt{dec}_2) = 1 \text{ and } m_1 \neq m_2 \\ : \mathtt{com}, m_1, m_2, \mathtt{dec}_1, \mathtt{dec}_2 \leftarrow \mathcal{A}(R, m, z)\}\end{array}\right] \leq \mathsf{negl}(\lambda)$$

Trapdoorness: *For any* PPT *adversary* $\mathcal{A}$ *there exists a negligible function* negl *such that:*

$$\left|\Pr[\mathsf{ExpCom}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z) = 1] - \Pr[\mathsf{ExpTrapdoor}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{ExpCom}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z)$ *and* $\mathsf{ExpTrapdoor}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z)$ *are the experiments defined in Figure 1*[10]*.*

$\mathsf{ExpCom}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z)$:	$\mathsf{ExpTrapdoor}_{\mathcal{A},\mathsf{TC}}(1^\lambda, z)$:
1. On input 1^λ and z, $\mathcal{A}$ outputs (aux, m).	1. On input 1^λ and z, $\mathcal{A}$ outputs (aux, m)
2. $R \xleftarrow{\$} \{0,1\}^\lambda$, $(\mathtt{com}, \mathtt{dec}) \leftarrow \mathsf{Com}(m; R)$.	2. $r \xleftarrow{\$} \{0,1\}^\lambda$, $\widehat{\mathtt{com}} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^\ell; r)$ and $\widehat{\mathtt{dec}} \leftarrow \mathsf{Eq}(\widehat{com}, r, m)$.
3. $\mathcal{A}$ on input $(\mathtt{com}, \mathtt{dec}, z, \mathsf{aux})$ outputs a bit b and this is the output of the experiment.	3. $\mathcal{A}$ on input $(\widehat{\mathtt{com}}, \widehat{\mathtt{dec}}, z, \mathsf{aux})$ outputs a bit b and this is the output of the experiment.

Fig. 1: Trapdoor experiments

We do not include in our definition the hiding property since this is trivially implied by trapdorness.

3.6 Instantiation of Ambiguous Commitment Scheme

Let $\mathcal{C}_{\mathsf{NI}} = (\mathsf{NiCom}, \mathsf{NiDec})$ be a non-interactive statistically binding commitment scheme, we can construct an ambiguous commitment scheme as follows [KOS18].

- Com: On input $m \in \{0,1\}^\ell$,
  1. Let m_i be the i-th bit of m. For each $i \in [\ell]$, compute
  $$(\mathtt{com}_i^0, \mathtt{dec}_i^0) \xleftarrow{\$} \mathsf{NiCom}(m_i, R) \text{ and } (\mathtt{com}_i^1, \mathtt{dec}_i^1) \xleftarrow{\$} \mathsf{NiCom}(m_i, R),$$
  sample $d_i \xleftarrow{\$} \{0,1\}$ and set $\mathtt{dec}_i^\star = \mathtt{dec}_i^{d_i}$.
  2. Set $\mathtt{com} = \{\mathtt{com}_i^b\}_{i\in[n], b\in\{0,1\}}$ and $\mathtt{dec} = \{\mathtt{dec}_i^\star\}_{i\in[n]}$.
- Dec: On input $(\mathtt{com}, m, \mathtt{dec})$,
  1. Parse $\mathtt{com}$ as $\{\mathtt{com}_i^b\}_{i\in[n],b\in\{0,1\}}$ and $\mathtt{dec}$ as $\{\mathtt{dec}_i^\star\}_{i\in[n]}$.
  2. If, for all $i \in [\ell]$, exists $d_i \in \{0,1\}$ such that $\mathsf{NiDec}(\mathtt{com}_i^{d_i}, m_i, \mathtt{dec}_i^\star) = 1$ then return 1, else return 0.
- $\mathsf{Com}^{\mathsf{eq}}$: On input 1^λ and randomness R, use R as the follows.
  1. For each $i \in [\ell]$, sample a random bit d_i and compute
  $$(\mathtt{com}_i^0, \mathtt{dec}_i^0) \xleftarrow{\$} \mathsf{NiCom}(d_i, R) \text{ and } (\mathtt{com}_i^1, \mathtt{dec}_i^1) \xleftarrow{\$} \mathsf{NiCom}(1 - d_i, R).$$
  2. Return $\{\mathtt{com}_i^b\}_{i\in[n],b\in\{0,1\}}$.

[10] We assume wlog that $\mathcal{A}$ is stateful.

- Eq: On input $(\mathtt{com}, m, R)$,
  1. Use the same randomness R used in $\mathsf{Com}^{\mathsf{eq}}$ to recompute $\mathsf{Com}^{\mathsf{eq}}$ and obtain $\{\mathtt{com}_i^b, \mathtt{dec}_i^b\}_{i\in[\ell],b_i\in\{0,1\}}$ and $\{d_i\}_{i\in[\lambda]}$.
  2. For each $i \in [\ell]$, if $m_i = d_i$ then set $\mathtt{dec}_i^\star \leftarrow \mathtt{dec}_i^0$, else set $\mathtt{dec}_i^\star \leftarrow \mathtt{dec}_i^1$.
  3. Return $\mathtt{dec} = \{\mathtt{dec}_i^\star\}_{i\in[n]}$.

Theorem 1. *The protocol described above is an ambiguous commitment scheme.*

3.7 One-of-Two Binding Commitments

We propose a formal definition of the *one-of-two binding commitments* proposed by Khurana et al. in [KOS18]. A one-of-two binding commitment is a three-round interactive protocol Π_{comWI} executed between a prover $\mathcal{P}_{\mathsf{comWI}}$ and a verifier $\mathcal{V}_{\mathsf{comWI}}$. Informally, it works as follows: the prover generates two commitments in the first round, and sends their opening third round; in parallel, the prover performs a WI proof that guarantees that at least one of the two commitments is binding. Moreover, the prover can equivocate the non-binding commitment to any value he likes. In [KOS18] the authors propose a one-of-two binding commitment scheme that makes black-box use of one-to-one OWFs. We propose a formal definition of the properties held by a one-of-two binding commitment scheme. We assume the prover and verifier algorithms are stateful in the following definitions.

Definition 6 (One-of-Two Binding Commitments). *A commitment is one-of-two binding if the following properties hold.*
Correctness.

- *The prover $\mathcal{P}_{\mathsf{comWI}}$ on input 1^λ, the message $m_b \in \{0,1\}^\lambda$, and a bit b returns π_1^{comWI}.*
- *The verifier on input 1^λ and π_1^{comWI} samples a random $\pi_2^{\mathsf{comWI}} \xleftarrow{\$} \{0,1\}^\lambda$ and returns it.*
- *The prover on input π_2^{comWI} and a message $m_{1-b} \in \{0,1\}^\lambda$ computes π_3^{comWI} and returns $(\pi_3^{\mathsf{comWI}}, m_0, m_1)$*
- *The verifier on input $(\pi_1^{\mathsf{comWI}}, \pi_2^{\mathsf{comWI}}, \pi_3^{\mathsf{comWI}}, m_0, m_1)$ returns $d \in \{0,1\}$, where $d = 1$ denotes that the verifier accepts, and 0 that he rejects.*

Binding. *For any* PPT *adversary $\mathcal{A}$, we have that the following holds. Let $\tau = (\pi_1^{\mathsf{comWI}}, \pi_2^{\mathsf{comWI}})$ be the first two rounds generated during the execution of Π_{comWI} by an honest receiver $\mathcal{V}_{\mathsf{comWI}}$ and the stateful adversarial prover $\mathcal{A}(1^\lambda)$. We have that*

$$\Pr[(\pi_3^{\mathsf{comWI}}, m_0, m_1, \overline{\pi}_3^{\mathsf{comWI}}, \overline{m}_0, \overline{m}_1) \leftarrow \mathcal{A}(1^\lambda) |\ \mathcal{V}_{\mathsf{comWI}}(\tau, \pi_3^{\mathsf{comWI}}, m_0, m_1) = 1 \wedge \mathcal{V}_{\mathsf{comWI}}(\tau, \overline{\pi}_3^{\mathsf{comWI}}, \overline{m}_0, \overline{m}_1) = 1 \ \wedge\ m_0 \neq \overline{m}_0 \text{ and } m_1 \neq \overline{m}_1] = \mathsf{negl}(\lambda)$$

Equivocability. *For any adversary $\mathcal{A}$ and any $m_0, m_1 \in \{0,1\}^\lambda$ we have that $\left|\Pr[b' = b] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$ in the following game.*
$\mathsf{ExpEq}_{\mathcal{A},\Pi}(1^\lambda, b, m_0, m_1)$:

1. *The challenger sends $\pi_1^{\mathsf{comWI}} \leftarrow \mathcal{P}_{\mathsf{comWI}}(1^\lambda, m_b, b)$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ sends π_2^{comWI} to the challenger*
3. *The challenger sends $\pi_3^{\mathsf{comWI}} \leftarrow \mathcal{P}_{\mathsf{comWI}}(\pi_2^{\mathsf{comWI}}, m_{1-b})$ to $\mathcal{A}$.*
4. *The adversary $\mathcal{A}$ outputs a bit b'.*

3.8 Proof Systems

Definition 7 (Soundness). *A pair of* PPT *interactive algorithms* $\Pi = (\mathcal{P}, \mathcal{V})$ *constitute a* sound system *for an* $\mathcal{NP}$*-language* L *that is associated with the relation* Rel*, if the following conditions hold:*

- Completeness: *For every* $x \in L$ *and* w *such that* $\mathsf{Rel}(x, w) = 1$*, it holds that* $\mathcal{V}$ *accepts the proof with probability* 1*.*
- Soundness: *For every* PPT *algorithm* $\mathcal{P}^\star$ *there exists a negligible function* negl *such that for every* $x \notin L$ *and every auxiliary input* z*:*

$$\Pr\left[\langle \mathcal{P}^\star(z), \mathcal{V}\rangle(x) = 1\right] \leq \mathsf{negl}(|x|).$$

We recall that a proof 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 formally defined in [CPS+16].

For a protocol that enjoys delayed-input completeness, we consider also the notion of *adaptive-input proof system.* That is, the soundness holds against a stronger adversary $\mathcal{P}^\star$ that can choose the statement to be proven in the last round of the interaction with $\mathcal{V}$.

An interactive protocol $\Pi = (\mathcal{P}, \mathcal{V})$ is *public coin* if, at every round, $\mathcal{V}$ simply tosses a predetermined number of coins (i.e., a random challenge) and sends the outcome to the prover. Moreover we say that the transcript τ of an execution $b = \langle \mathcal{P}(w), \mathcal{V}\rangle(x)$ is *accepting* if $b = 1$.

A *3-round protocol* $\Pi = (\mathcal{P}, \mathcal{V})$ for a relation Rel is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ on common input x and private input w of $\mathcal{P}$ such that $\mathsf{Rel}(x, w) = 1$. More precisely, a 3-round protocol $\Pi = (\mathcal{P}, \mathcal{V})$ works as follow:

– $\mathcal{P}$, on input a security parameter λ, x and w, computes the first message π^1 with an auxiliary information aux, and sends π^1 to $\mathcal{V}$.
– $\mathcal{V}$ sends a random challenge π^2 to $\mathcal{P}$.
– Upon receiving π^2, $\mathcal{P}$ on input π^2 and aux computes and sends π^3 to $\mathcal{V}$.

At the end of the protocol, $\mathcal{V}$ decides to accept or reject based on the messages that they have seen (i.e., x, π^1, π^2, π^3). We usually denote the message π^2 sent by $\mathcal{V}$ as a *challenge.*

We recall the following definitions.

Definition 8 (Special Honest Verifier Zero-knowledge (SHVZK)). *A 3-round protocol* $\Pi = (\mathcal{P}, \mathcal{V})$ *as defined above, is* special honest-verifier zero-knowledge *(SHVZK) if there exists a* PPT *algorithm* Sim *that for any* $x \in L$*, where* L *is an* $\mathcal{NP}$*-language, security parameter* λ *and any challenge* π^2 *works as follow:* $(\pi^1, \pi^3) \leftarrow \mathsf{Sim}(1^\lambda, x, \pi^2)$*. Furthermore, the distribution of the output of* Sim *is computationally indistinguishable from the distribution of a transcript obtained when* $\mathcal{V}$ *sends* π^2 *as challenge and* $\mathcal{P}$ *runs on common input* x *and any* w *such that* $\mathsf{Rel}(x, w) = 1$*.*

Definition 9 (Adaptive-input SHVZK). *A delayed-input 3-round protocol* $\Pi = (\mathcal{P}, \mathcal{V})$ *for relation* Rel *satisfies* adaptive-input special honest-verifier zero-knowledge (AI-SHVZK) *if there exists a* PPT *simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *such that for all* PPT *adversaries* $\mathcal{A}$ *and for all challenges* π^2 *there is a negligible function* negl *for which* $\left|\Pr[b' = b] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$ *in the following game.*

$\mathsf{ExpAISHVZK}_{\mathcal{A},\Pi}(1^\lambda, b, \pi^2)$:

1. *The challenger sends* π^1 *to* $\mathcal{A}$*, where:*
   - *If* $b = 0$, $(\pi^1, \mathsf{aux}) \leftarrow \mathcal{P}(1^\lambda, 1^m)$*, with* $m = |x|$
   - *Else, if* $b = 1$, $(\pi^1, \mathsf{aux}) \leftarrow \mathsf{Sim}_0(1^\lambda, 1^m, \pi^2)$
2. $\mathcal{A}$ *sends* (x, w) *to the challenger.*
   - *If* $(x, w) \in \mathsf{Rel}$*, the challenger sends* π^3 *to* $\mathcal{A}$*, where:*
     - *If* $b = 0$, $\pi^3 \leftarrow \mathcal{P}(x, w, \mathsf{aux}, \mathsf{aux}, \pi^2)$
     - *Else, if* $b = 1$, $\pi^3 \leftarrow \mathsf{Sim}_1(x, \mathsf{aux})$
   - *Else, the challenger sends* $\pi^3 = \perp$ *to* $\mathcal{A}$
3. *The adversary* $\mathcal{A}$ *outputs a bit* b'*.*

3.9 Commit-and-Prove

Definition 10 (Commit-and-Prove Proof of Knowledge). *We propose a revisit of the definition proposed in [KOS18]. A commit-and-prove proof of knowledge* $(\mathcal{P}(m, |x|), \mathcal{V}(|x|))$ *is an interactive protocol between a prover* $\mathcal{P}$ *and verifier* $\mathcal{V}$*. It consists of two phases, a commit phase-and-reveal phase.*

- *In the commit phase,* $\mathcal{P}$ *interacts with* $\mathcal{V}$ *to commit to a message* m*. It also proves that the message* m *satisfies some relation* Rel*, in other words it proves that* $\mathsf{Rel}(x, m) = 1$*, for same public statement* x *(that can be adaptively chosen by* $\mathcal{P}$ *in the last round of the commitment phase). Prover and verifier also store the (private) randomness used to generate the protocol messages denoted respectively by* $\mathtt{state}_\mathcal{P}$ *and* $\mathtt{state}_\mathcal{V}$*. At the end of the commitment phase,* $\mathcal{V}$ *outputs* 0 *or* 1*, where* 1 *denotes that* $\mathcal{V}$ *accepted the commit phase. We denote by* τ *the transcript obtained during the commitment phase:* $\tau \leftarrow \mathtt{CommitProve}\langle \mathcal{P}(|x|, m), \mathcal{V}(|x|)\rangle$*, where* τ *contains the statement* x *proven by the prover.*
- *Later, in the reveal phase, parties* $\mathcal{P}$ *and* $\mathcal{V}$ *possibly engage in another decommit phase, which we denote by* $\mathtt{Decommit}\langle \tau, \mathcal{P}(m, \mathtt{state}_\mathcal{P}), \mathcal{V}(\mathtt{state}_\mathcal{V})\rangle$*, at the end of which* $\mathcal{V}$ *outputs* $\perp$ *or* $\tilde{m} \in \{0,1\}^\ell$*.*

We require the protocol to satisfy the following conditions.

Completeness. *If* $\mathcal{P}$ *and* $\mathcal{V}$ *honestly follow the protocol, then the probability that* $\mathcal{V}$ *accepts the proof is* 1*.*

Proof of Knowledge.[11] *There exists a* PPT *oracle algorithm* E *that given oracle access to a corrupted* PPT *prover* $\mathcal{P}^\star$ *that provides an accepting transcript* $\tau \leftarrow \mathtt{CommitProve}\langle \mathcal{P}^\star, \mathcal{V}(|x|)\rangle$ *with some non-negligible probability* $p(\lambda)$ *outputs* $\tilde{m} \neq \perp$ *with non-negligible probability* $q(|x|)$ *(and returns always* $\tilde{m} = \perp$ *with probability* $1 - q(\lambda)$*) such that the following properties are satisfied:*

1. $\mathsf{Rel}(x, \tilde{m}) = 1$.
2. $\Pr\left[m \leftarrow \mathtt{Decommit}\langle \tau, \mathcal{P}^\star, \mathcal{V}(\mathtt{state}_\mathcal{V})\rangle \ \wedge \ m \neq \tilde{m}\right] \leq \mathsf{negl}(\lambda)$.

[11] Our notion differs from PoK notions introduced in previous works as it requires the existence of a PPT extractor (instead of an expected PPT one), and it requires the extractor to be successful only with a non-negligible probability. We consider this weaker version of PoK because this is sufficient to prove the security of our non-malleable commitment scheme.

3.10 MPC Definitions

Definition 11 (Correctness). *We say that a protocol $\Pi = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$ realizes a deterministic n-party functionality $f(x, w_1, \ldots, w_n)$ with* perfect (resp. statistical) correctness *if for all inputs $(x, w_1, \ldots, w_n)$ the probability that the output $\mathtt{out}_i$ of some party $\mathtt{P}_i$ is different from the output of f is 0 (resp. negligible), where the probability is over the independent choices of the random tapes $r_1, \ldots, r_n$.*

Definition 12 (t_p-Privacy). *Let $1 \leq t_p < n$, we say that the protocol $\Pi = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$ realizes f with* perfect t_p-privacy *if for any input $(x, w_1, \ldots, w_n)$ and for all $A \subseteq [n]$, where $|A| \leq t_p$, there exists a* PPT *simulator* Sim *such that, $\mathsf{Sim}(A, x, \{w_i\}_{i \in A}, f_A(x, w_1, \ldots, w_n))$ is identically distributed to the joint view $\mathsf{view}_A(x, w_1, \ldots, w_n) = \{\mathsf{view}_i\}_{i \in A}$ of parties in A.*
We will speak about statistical (resp. computational) t_p-privacy if the two distributions $\mathsf{view}_A(x, w_1, \ldots, w_n)$ and $\mathsf{Sim}(A, x, \{w_i\}_{i \in A}, f_A(x, w_1, \ldots, w_n))$ are statistically (resp. computationally) indistinguishable.

We will need the following definitions from [IKOS07].

Definition 13 (Consistent views). *We say that a pair of views $\mathsf{view}_i, \mathsf{view}_j$, $i, j \in [n]$, computed w.r.t. the randomness r_i, r_j, are* consistent *(with respect to the protocol Π and some public input x) if the outgoing messages implicit in view_i, are identical to the incoming messages reported in view_j and vice versa.*

Lemma 1 (Local vs. global consistency). *Let Π be an n-party protocol as above and x be a public input. Let $\mathsf{view}_1, \ldots \mathsf{view}_n$ be an n-tuple of (possibly incorrect) views. Then all pairs of views $\mathsf{view}_i, \mathsf{view}_j$ are consistent with respect to Π and x if and only if there exists an honest execution of Π with public input x (and some choice of private inputs w_i and random tapes r_i) in which view_i is the view generated using the code of $\mathtt{P}_i$ using randomness r_i for every $i \in [n]$.*

3.11 Verifiable Secret Sharing (VSS)

A verifiable secret sharing (VSS) scheme [CGMA85] is a two-phase protocol carried out among $n+1$ parties. In the first step, a special party, also referred to as the *dealer*, shares a secret among all the other n parties, referred to as *share-holders*, at most t of whom may be corrupt; in the second step, parties reconstruct the secret. While in standard secret-sharing schemes the dealer is assumed to be honest, in VSS schemes also the dealer can be corrupt. Loosely speaking, if the dealer is honest, then no information about the dealer's secret is revealed to the t corrupt parties by the end of the sharing phase; moreover, by the end of the sharing phase even a dishonest dealer is committed to some value that will be recovered by the honest parties in the reconstruction phase. Furthermore, if the dealer is honest then this committed value must be identical to the dealer's initial input.

Definition 14 (Verifiable Secret Sharing [CGMA85, CLP20]). *An $(n+1, t)$-perfectly secure Verifiable Secret Sharing (VSS) scheme Π^σ consists of a pair of protocols* $(\mathsf{Share}, \mathsf{Recon})$ *that implement respectively the sharing and reconstruction phases as follows.*

- *Sharing Phase* (Share). *Party $\mathtt{P}_{n+1}$ (the dealer) runs on input a secret s and randomness r_{n+1}, while any other party $\mathtt{P}_i$, $i \in [n]$, runs on input a randomness r_i. During this phase parties can send (both private and broadcast) messages in multiple rounds. We will indicate with view_i the view that $\mathtt{P}_i$ obtains at the end of sharing phase, and with $(\mathsf{view}_1, \ldots, \mathsf{view}_n) = \mathsf{Share}(s, r_1, \ldots, r_n, r_{n+1})$ the process described above.*

- *Reconstruction Phase* (Recon). *Each shareholder sends its view* view_i, $i \in [n]$, *of the sharing phase to each other party, and on input the views of all parties (that might include corrupt or empty views) each party outputs a reconstruction of the secret* s. *All computations performed by honest parties are efficient.*

The following security properties hold even if an unbounded adversary corrupts up to t *parties (hence,* t *parties can deviate from the above procedures).*

Commitment. *If the dealer is dishonest then one of the following two cases happen: 1) during the sharing phase honest parties disqualify the dealer, therefore they output a special value* $\perp$ *and will refuse to run the reconstruction phase; 2) during the sharing phase honest parties do not disqualify the dealer, therefore such a phase determines a unique value* s^*, *that belongs to the set of possible legal values that does not include* $\perp$, *which will be reconstructed by the honest parties during the reconstruction phase.*

Secrecy. *If the dealer is honest, then the adversary's view during the sharing phase reveals no information about* s. *More formally, the adversary's view is identically distributed under all different values of* s.

Perfect Correctness. *If the dealer is honest throughout the protocols then each honest party will output the shared secret* s *at the end of protocol* Recon *with probability* 1.

Assuming a broadcast channel, perfectly-secure $(n+1, \lfloor n/4 \rfloor)$-VSS scheme are implemented in [GIKR01].

4 Non-Malleable HVZK with respect to Commitment

In this section, we introduce the new notion of non-malleable HVZK with respect to commitment (NMZKC). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a proof system, and $\Pi_{\mathtt{com}}$ be a (potentially interactive) commitment scheme. We consider a scenario where a man-in-the-middle adversary $\mathcal{A}$ interacts in the left session with the prover of Π (hence, $\mathcal{A}$ acts as the verifier for Π), and in the right session $\mathcal{A}$ acts as the sender for $\Pi_{\mathtt{com}}$ against an honest receiver. the formal definition of NMZKC follows, and we refer to the introductory section of the paper for an informal discussion about this definition. Let $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ be the adaptive-input HVZK simulator for Π, we define the experiment $\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, b, c)$.

$\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, b, c)$: In the right session, interact with $\mathcal{A}$ as the receiver of $\Pi_{\mathtt{com}}$. In the left session, act as follows.

1. Set $\pi_2 \leftarrow c$ and send π^1 to $\mathcal{A}$, where:
   - If $b = 0$, $(\pi^1, \mathsf{aux}) \xleftarrow{\$} \mathcal{P}(1^\lambda, 1^m)$, with $m = |x|$
   - If $b = 1$, $(\pi^1, \mathsf{aux}) \xleftarrow{\$} \mathsf{Sim}_0(1^\lambda, 1^m, \pi^2)$
2. Upon receiving (x, w) from $\mathcal{A}$ in the left session do the following
   - If $(x, w) \in \mathsf{Rel}$, the experiment sends π^3 to $\mathcal{A}$ in the left session where:
     - If $b = 0$, $\pi^3 \leftarrow \mathcal{P}(x, w, \mathsf{aux}, \pi^2)$
     - Else, if $b = 1$, $\pi^3 \xleftarrow{\$} \mathsf{Sim}_1(x, \mathsf{aux})$
   - Else, the experiment sets $\pi^3 \leftarrow \perp$
3. Set the output of the experiment as the output of $\mathcal{A}$ and its view.

Definition 15 (NMZKC). *Let $\Pi_{\mathtt{com}}$ be a commitment scheme. We say that an adaptive-input HVZK proof system Π, with challenge space $\mathcal{C}$, is a* non-malleable HVZK with respect to commitment for $\Pi_{\mathtt{com}}$ *if there exists a* PPT *simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *such that, for all* PPT *adversary $\mathcal{A}$, the following two distributions are indistinghuishable:*

$$\{\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, 0, c), m_0\}_{\lambda\in\mathbb{N},c\in\mathcal{C}}$$

$$\{\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, 1, c), m_1\}_{\lambda\in\mathbb{N},c\in\mathcal{C}},$$

where $\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, b, c)$ *is the experiment described above and m_b, with $b \leftarrow \{0,1\}$, is the message committed in the right session of* $\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, b, c)$ *by $\mathcal{A}$.*

We note that non-malleable HVZK with respect to commitment property is parallel composable w.r.t. multiple left sessions. The proof would follow via standard hybrid arguments. We also consider a weaker notion of NMZKC, in which the adversary needs to pick the witness at the on-set of the experiment (but the statement is fully adaptive on the first round). We refer to this notion as *adaptive-theorem* NMZKC.

5 Robust Security for MPC

In this work, we consider MPC protocols $\Pi = \Pi^{\mathtt{off},\mathtt{on}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$, among n parties $\mathtt{P}_1, \ldots, \mathtt{P}_n$, that are composed of two sub-protocols $\Pi^{\mathtt{off}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$ and $\Pi^{\mathtt{on}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$, where the execution $\Pi^{\mathtt{off}}$ does not require parties' private inputs, namely $\Pi^{\mathtt{off}}$ is *input independent*. If each party $\mathtt{P}_i$, for $i \in [n]$, runs Π honestly, then the execution of Π is called an *honest execution*. A view view_i of a party $\mathtt{P}_i$ is composed by its private input w_i, randomness r_i, and transcript τ_i, where τ_i is given by the set of messages received and sent by party $\mathtt{P}_i$ during the execution of the MPC protocol Π. We denote the view of the offline and of the online phase for a party P_i with $\mathsf{view}_i^{\mathtt{off}}$ and $\mathsf{view}_i^{\mathtt{on}}$ respectively.

In the rest of the paper, we consider MPC protocols where all parties share a public input x, and each party $\mathtt{P}_i$ additionally holds a local private input w_i and random tape r_i. We consider protocols $\Pi^{\mathtt{off},\mathtt{on}}$ which securely realize an n-party functionality f. The output $y = f(x, w_1, \ldots, w_n)$ can be computed from any $\mathsf{view}_i = (\mathsf{view}_i^{\mathtt{off}}, \mathsf{view}_i^{\mathtt{on}})$, i.e., $y = \Pi_f^{\mathtt{off},\mathtt{on}}(\mathsf{view}_i) = \mathtt{out}_i$, for each $i \in [n]$.

Looking ahead, in our delayed-input protocol the prover, while committed to $\mathsf{view}_1^{\mathtt{off}}, \ldots, \mathsf{view}_n^{\mathtt{off}}$, is allowed to generate the online views $\mathsf{view}_1^{\mathtt{on}}, \ldots, \mathsf{view}_n^{\mathtt{on}}$ only when it received (x, w), and after it is given any eventual random inputs and the set of k parties/views it will need to open. This means that a malicious prover $\mathcal{P}$ might arbitrarily create inconsistent views $\mathsf{view}_{i_1}^{\mathtt{on}}, \ldots, \mathsf{view}_{i_{n-k}}^{\mathtt{on}}$ that will not be opened, easily making all outputs to be incorrect without being caught. For this reason we need an underlying MPC protocol with strong security requirements and introduce the following definition of *robustness*.

Despite the name, this notion is different from the definition of robustness that was given in [IKOS07] to generalize the definition of correctness in case of malicious adversaries.

Roughly, an MPC protocol $\Pi = \Pi^{\mathtt{off},\mathtt{on}}$ is said to be robust if, given two subsets $A, H \subset [n]$, with $|H| = n - |A|$, and a correct execution of $\Pi^{\mathtt{off}}$, the output $\mathtt{out}_j$ of some $\mathtt{P}_j$, with $j \in A$, obtained by running the protocol on input $(x, (w_i)_{i\in A}, (w_i)_{i\in H})$ and using some arbitrary randomness r'_j, is not $\perp$ then $\mathtt{out}_j = y$, where $y = \Pi_f^{\mathtt{off},\mathtt{on}}(\mathsf{view}_i)$, $\forall i \in H$. Note that our definition specifically assumes

an MPC protocol $\Pi^{\mathtt{on},\mathtt{off}}$ in the pre-processing model with a correctly executed $\Pi^{\mathtt{off}}$ and requires that every unbounded adversary $\mathcal{A}$ cannot make the parties in A output a result inconsistent with the views of honest parties. The formal definition of robustness follows.

Definition 16 (Robustness). *Let $\Pi^{\mathtt{off},\mathtt{on}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$ be as above. Let $A \subset [n]$ and $H = [n] - A$. Let us denote by* view *the view $\{\mathsf{view}_i = (\mathsf{view}_i^{\mathtt{off}}, \mathsf{view}_i^{\mathtt{on}})\}_{i\in H}, \{\widetilde{\mathsf{view}}_i = (\widetilde{\mathsf{view}}_i^{\mathtt{off}}, \widetilde{\mathsf{view}}_i^{\mathtt{on}})\}_{i\in A}$, such that:*

- $\widetilde{\mathsf{view}}_i^{\mathtt{off}}$ *and* $\widetilde{\mathsf{view}}_i^{\mathtt{on}}$ *are the views generated by running the code of $\mathtt{P}_i$ for $\Pi^{\mathtt{off}}$ and $\Pi^{\mathtt{on}}$ on input (x, w_i), respectively, with some arbitrary randomness $r'_i \in \{0,1\}^\lambda$, for each $i \in A$;*
- $\mathsf{view}_i^{\mathtt{off}}$ *is the view generated running the code of party $\mathtt{P}_i$ for $\Pi^{\mathtt{off}}$ with some arbitrary randomness $r'_i \in \{0,1\}^\lambda$, for each $i \in H$;*
- $\mathsf{view}_i^{\mathtt{on}} \in \{0,1\}^*$*, for each $i \in H$.*

We say that $\Pi^{\mathtt{off},\mathtt{on}}$ realizes a deterministic n-party functionality $f(x, w_1, \ldots, w_n)$ with robustness *if for any A and H, such that $H = \{i_1, \ldots, i_{n-t}\}$ and $A = \{j_1, \ldots, j_t\}$, the following holds: if, for each $j_k \in A$, party $\mathtt{P}_{j_k}$, on input randomness r_{j_k} and (x, w_{j_k}), outputs $\mathtt{out}_{j_k} = F \neq \perp$ with respect to the view* view*, then $F = f_A(x, w_{i_1}, \ldots, w_{i_{n-t}})$, for some $w_{i_1}, \ldots, w_{i_{n-t}}$ with $\{i_1, \ldots, i_{n-t}\} = H$, where f_A is the function evaluated on n inputs where the inputs in positions $A = \{j_1, \ldots, j_t\}$ are $w_{j_1}, \ldots, w_{j_t}$.*

Intuitively, the above definition says that as long as $\Pi^{\mathtt{off}}$ is correct (concretely this can be achieved instantiating $\Pi^{\mathtt{off}}$ with a malicious secure protocol) and the online phase $\Pi^{\mathtt{on}}$ is a deterministic function of the offline phase, then Π is robust. Notice the definition of robustness is independent of the number of corruptions supported by Π and it can be achieved both with an honest and dishonest majority. In Section 10, we show a concrete instantiation of a robust MPC protocol.

6 Our Delayed-Input MPC-in-the-Head Protocol $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$

Let L be an $\mathcal{NP}$-language and Rel be the corresponding $\mathcal{NP}$-relation. Let f be an $(n+1)$-argument function, with $n > 2$, corresponding to Rel, i.e., $f(x, w_1, \ldots, w_n) = \mathsf{Rel}(x, w_1 \oplus \cdots \oplus w_n)$. Our protocol, $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$, for the $\mathcal{NP}$-relation Rel makes use of the following tools:

- A t_p-private MPC protocol $\Pi^{\mathtt{off},\mathtt{on}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$ that realizes f with robustness (Definition 16).
- An ambiguous commitment scheme $\Pi_{\mathtt{com}} = (\mathsf{Com}, \mathsf{Dec}, \mathsf{Com}^{\mathsf{eq}}, \mathsf{Eq})$ as in Definition 5.

A complete description of $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ for the $\mathcal{NP}$-relation Rel can be found in Figure 2. At a high level, given an MPC protocol $\Pi^{\mathtt{off},\mathtt{on}}$, as specified above, $\mathcal{P}_{\mathsf{AI}}$ starts by emulating $\Pi^{\mathtt{off}}$ in its head. In particular, it generates n views $\mathsf{view}_i^{\mathtt{off}}, i \in [n]$, corresponding to the n virtual parties and separately commits to these views using an ambiguous commitment scheme $\Pi_{\mathtt{com}}$. This is done by sampling c random values $\{\mathsf{view}_{(i,j)}^{\mathtt{off}}\}_{j\in[c]}$, for each $i \in [n]$, such that $\mathsf{view}_i^{\mathtt{off}} = \bigoplus_{j\in[c]} \mathsf{view}_{i,j}^{\mathtt{off}}$, and computing $\{(\mathtt{com}_{(i,j)}, \mathtt{dec}_{(i,j)}) \leftarrow \mathsf{Com}(\mathsf{view}_{(i,j)}^{\mathtt{off}}; R_{(i,j)})\}_{j\in[c]}$. Notice here $c \geq 2$ is a small integer. This will allow the verifier to check that the commitments are correctly generated and $\Pi^{\mathtt{off}}$ is honestly executed; moreover, it will be crucial to prove adaptive-input SHVZK, as we will see later.

The prover sends the first message π^1, given by the concatenation of all the commitments, to $\mathcal{V}$ which replies with the challenge π^2, i.e., a set of random indices $I = \{i_1, \ldots, i_k\} \subset [n]$ with $k \leq t_p$, and one index $q_{i_j} \in [c]$ for each $i \in I$.

COMMON INPUTS: At the beginning of the third round both $\mathcal{P}_{\mathsf{AI}}$ and $\mathcal{V}_{\mathsf{AI}}$ gets x, while the parameters k, c, n (which are small constants) and $k < t_p$ are specified when the protocol starts.
PRIVATE INPUT: At the beginning of the third round $\mathcal{P}_{\mathsf{AI}}$ gets a random n-out-of-n secret sharing of the witness $w = w_1 \oplus \cdots \oplus w_n$.

Round 1. $\mathcal{P}_{\mathsf{AI}}$ computes the following steps.

1. Run $\Pi^{\mathtt{off}}$ "in its head" (by choosing uniform random coins r_i for each party) to generate the transcript of each party $\mathtt{P}_i$. Let $\mathsf{view}_i^{\mathtt{off}}$ denote the view of $\mathtt{P}_i$ in the execution of $\Pi^{\mathtt{off}}$.
2. For each $i \in [n]$, choose c random values $\{\mathsf{view}_{(i,j)}^{\mathtt{off}}\}_{j\in[c]}$ such that $\mathsf{view}_i^{\mathtt{off}} = \mathsf{view}_{(i,1)}^{\mathtt{off}} \oplus \mathsf{view}_{(i,2)}^{\mathtt{off}}, \ldots, \oplus\, \mathsf{view}_{(i,c)}^{\mathtt{off}}$.
3. For each $i \in [n]$, compute $\{(\mathtt{com}_{(i,j)}, \mathtt{dec}_{(i,j)}) \leftarrow \mathsf{Com}(\mathsf{view}_{(i,j)}^{\mathtt{off}}; R_{(i,j)})\}_{j\in[c]}$.
4. Send $\{\mathtt{com}_{(1,j)}, \ldots, \mathtt{com}_{(n,j)}\}_{j\in[c]}$ to $\mathcal{V}_{\mathsf{AI}}$.

Round 2. $\mathcal{V}_{\mathsf{AI}}$ chooses a random a subset of distinct indices $I = \{i_1, \ldots, i_k\} \subset [n]$, with $|I| = k \leq t_p$; and for each index i_j it chooses a random value $q_{i_j} \in [c]$.
$\mathcal{V}_{\mathsf{AI}}$ sends $(I, q_{i_1}, \ldots, q_{i_k})$ to $\mathcal{P}_{\mathsf{AI}}$.

Round 3. Upon receiving $(x, (w_1 \oplus \cdots \oplus w_n))$, where $w = w_1 \oplus \cdots \oplus w_n$ s.t. $\mathsf{Rel}(x, w) = 1$, $\mathcal{P}_{\mathsf{AI}}$ computes the following steps:

1. Simulate the behaviour of the party $\mathtt{P}_i$ while running $\Pi^{\mathtt{on}}$ on input r_i, x, w_i.
   For each $i_j \in I$, let view_{i_j} be the view of $\mathtt{P}_{i_j}$ in the execution of Π which is composed of $\mathsf{view}_{i_j}^{\mathtt{off}}$ and $\mathsf{view}_{i_j}^{\mathtt{on}}$.
2. Let $C_{i_j} = \{1, \ldots, c\} \setminus \{q_{i_j}\}$. For each $i_j \in I$, send to $\mathcal{V}_{\mathsf{AI}}$ the following: $\Big(\{(\mathsf{view}_{(i_j,l)}^{\mathtt{off}}, \mathtt{dec}_{(i_j,l)})\}_{l\in C_{i_j}},\ (\mathsf{view}_{(i_j,q_{i_j})}^{\mathtt{off}}, R_{(i_j,q_{i_j})}),\ \mathsf{view}_{i_j}^{\mathtt{on}}\Big)$.

Verification step. $\mathcal{V}_{\mathsf{AI}}$ outputs 1 if and only if all the following checks pass.

1. For $i_j \in I$ check that
   - $\mathsf{Dec}(\mathtt{com}_{(i_j,l)}, \mathsf{view}_{(i_j,l)}^{\mathtt{off}}, \mathtt{dec}_{(i_j,l)}) = 1$, for all $l \in C_{i_j}$
   - $\mathsf{Com}(\mathsf{view}_{(i_j,q_{i_j})}^{\mathtt{off}}; R_{(i_j,q_{i_j})}) = \mathtt{com}_{(i_j,q_{i_j})}$.
2. The output of $\mathtt{P}_{i_j}$ is $\neq \perp$, for each $i_j \in I$.
3. The views $\mathsf{view}_{i_1}, \ldots, \mathsf{view}_{i_k}$ are consistent, where $\mathsf{view}_{i_j}^{\mathtt{off}} = \bigoplus_{l\in[c]} \mathsf{view}_{(i_j,l)}^{\mathtt{off}}$

Fig. 2: $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$

In the last round, both $\mathcal{P}$ and $\mathcal{V}$ receive the theorem x, while $\mathcal{P}$ also receives w. in the last round, $\mathcal{P}$ first completes the emulation of the MPC protocol, producing all the online views $\mathsf{view}_i^{\mathtt{on}}, i \in [n]$; secondly, it sends $\mathsf{view}_i^{\mathtt{on}}, i \in I$, and opens the corresponding commitments in π^1 as follows. The commitments corresponding to the indices q_{i_j} in π^2 are opened in a "binding way", by sending $\mathsf{view}_{i_j,q_{i_j}}^{\mathtt{off}}$ and $R_{i_j,q_{i_j}}$, $i_j \in I$, and the remaining $c-1$ commitments, for each $i_j \in I$, are opened by sending the opening information $\mathtt{dec}_{i_j,q}$, along with $\mathsf{view}_{i_j,q}^{\mathtt{off}}$, for each $q \in \{1, \ldots, c\} \setminus q_{i_j}$.

Finally, the verifier checks all the commitments. It verifies that all the parties in I output 1 and that their views are consistent with each other. We finally note that our protocol can be parameterized to work with any n-out-of-n secret sharing scheme. Moreover, it would remain black-box in the use of the underlying cryptographic primitives as long the reconstruction phase of the secret sharing scheme does make any calls to a cryptographic primitive. We prove the following result.

Theorem 2. *If $\Pi^{\mathtt{off},\mathtt{on}}$ is an MPC protocol that realizes f (which is described above) with t_p-privacy and robustness, and $\Pi_{\mathtt{com}}$ is an ambiguous commitment scheme, let Π_{ComExt} be a 3-round extractable commitment scheme with a polynomial time extractor* Ext *that is successful with non-negligible probability, then Π_{AI} is a 3-round public-coin adaptive-input sound delayed-input protocol (with constant soundness error) for the $\mathcal{NP}$-relation* Rel *satisfying the property of non-malleable HVZK with respect to commitment for Π_{ComExt}.*

Correctness follows by inspection.

Adaptive-input soundness (Intuition). At a high level, we can see that soundness can be proved using the robustness property of the MPC protocol Π and the security properties of $\Pi_{\mathtt{com}}$. If all the offline views are correctly generated, then robustness ensures that a malicious prover will always get caught. Hence a malicious prover can succeed either if incorrect offline views are generated, or if some of the commitments are not computed in *binding mode*. We can argue that the probability of the adversary being caught in either of the two cases is noticeable.

Adaptive-input special honest-verifier zero-knowledge (Intuition). At a high level, the simulator $\mathsf{Sim} = (\mathsf{Sim}^0_{\mathsf{AI}}, \mathsf{Sim}^1_{\mathsf{AI}})$ works as follows. Let the challenge be $(I, q_{i_1}, \ldots, q_{i_k})$, and let $C_{i_j} = \{1, \ldots, c\} \setminus \{q_{i_j}\}$. For each $i_j \in I$, and each $l \in C_{i_j}$, $\mathsf{Sim}^0_{\mathsf{AI}}$ computes a random value $\mathsf{view}_{(i_j,l)}$. Then $\mathsf{Sim}^0_{\mathsf{AI}}$ generates the following commitments. For each $i_j \notin I$ and $q \in [c]$ set $\mathtt{com}_{(i_j,q)}$ as a commitment of the the all-zero string; for each $i_j \in I$ compute the commitment $\mathtt{com}_{(i_j,q_{i_j})}$ in binding mode, and for each $l \in C_{i_j}$ compute $\mathtt{com}_{(i_j,l)}$ in equivocal mode. These commitments constitute the simulated message π^1. In the second phase, when x is available, $\mathsf{Sim}^1_{\mathsf{AI}}$ uses the MPC simulator to obtain $(\mathsf{view}^{\mathtt{off}}_i, \mathsf{view}^{\mathtt{on}}_i), i \in [n]$. For each $i_j \in I$ and for each $l \in C_{i_j}$ compute $\mathsf{view}^{\mathtt{off}}_{i_j,l}$, such that $\mathsf{view}^{\mathtt{off}}_{i_j,q_{i_j}} = \mathsf{view}^{\mathtt{off}}_{i_j} \bigoplus_{l \in C_{i_j}} \mathsf{view}^{\mathtt{off}}_{i_j,l}$. Finally, for each $i_j \in I$, $l \in C_{i_j}$ equivocate the commitment $\mathtt{com}_{i_j,l}$ to $\mathsf{view}^{\mathtt{off}}_{i_j,l}$, and sends the openings of all the commitments to complete the third round. □

Lemma 2. *Let Π_{ComExt} be a 3-round extractable commitment scheme with a polynomial time extractor Ext, that extracts with non-negligible probability, then Π_{AI} is non-malleable HVZK with respect to commitment for Π_{ComExt}.*

Proof. To prove Lemma 2 we proceeds via hybrid experiments.

H_0: This hybrid corresponds to $\mathsf{ExpZK}_{\mathcal{A},\Pi,\Pi_{\mathtt{com}}}(1^\lambda, 0, I)$. H_0, in the left session, on input the challenge $(i_1, q_{i_1}), \ldots, (i_k, q_{i_k})$, $i_j \in I$, computes the 1st round π^1 of Π_{AI} using the honest procedure of $\mathcal{P}_{\mathsf{AI}}$ and sends it to $\mathcal{A}$.
Upon receiving x, w from $\mathcal{A}$, the experiment computes the 3rd round π^3 of Π_{AI} using the honest procedure of $\mathcal{P}_{\mathsf{AI}}$ w.r.t. x, w and send π^3 to $\mathcal{A}$.
H_0 acts as honest receiver in the right session.

H_2: This hybrid is described as the previous one except for how the commitments are computed. Formally, let $I = \{i_1, \ldots, i_k\}$ and C_{i_j} be the set of all indices in $[c] - \{q_{i_j}\}$, for each $j \in [k]$.
- Compute $\{\mathsf{view}^{\mathtt{off}}_{i,q}\}_{i \in [n], q \in [c]}$ as in the honest prover procedure.
- For each $i \notin I$ and $q \in [c]$, compute $(\mathtt{com}_{(i,q)}, \mathtt{dec}_{(i,q)}) \leftarrow \mathsf{Com}(\mathsf{view}^{\mathtt{off}}_{i_j,q}; R_{(i,q)})$. For each $i_j \in I$, compute
  * $(\mathtt{com}_{(i_j,q_{i_j})}, \mathtt{dec}_{(i,q_{i_j})}) \leftarrow \mathsf{Com}(\mathsf{view}^{\mathtt{off}}_{(i_j,q_{i_j})}; R_{(i_j,q_{i_j})})$.
  * For $l \in C_{i_j}$ compute $\mathtt{com}_{(i_j,l)} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^\lambda; R_{(i_j,l)})$.

This hybrid is indistinguishable from the previous one due to the trapdoor property of the ambiguous commitment scheme. Moreover, the distribution of the message committed in Π_{ComExt} does not change due to Claim 1.

H_2: This hybrid is described as the previous one except that, in the left session, the 1st round π^1 of Π_{AI} is computed as $\mathsf{Sim}^0_{\mathsf{AI}}$ does. All the commitments that will not be opened in the third round (we know the set of commitments since we know the challenge in advance by definition) are the commitment of 0^λ.
This hybrid is indistinguishable from the previous one due to the hiding property of the ambiguous commitment scheme. Moreover, the distribution of the message committed in Π_{ComExt} does not change due to the hiding property of $\Pi_{\mathtt{com}}$; the proof is similar to the one of Claim 2 below.

H_3: This hybrid is defined as the previous one except that, in the left session, the third round π^3 of Π_{AI} is computed using $\mathsf{Sim}^1_{\mathsf{AI}}$ w.r.t. theorem x specified by $\mathcal{A}$. This hybrid is indistinguishable from the previous one due to the t_p-privacy of $\Pi^{\mathtt{off},\mathtt{on}}$. Moreover, the distribution of the message committed in Π_{ComExt} does not change due to Claim 2.

Claim 1 *Let $\bar{p}$ be the probability that $\mathcal{A}$ changes the distribution of the committed messages between H_1 and H_0, then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

Proof. Suppose by contradiction that Claim 1 does not hold. Then it is possible to make a reduction that contradicts the trapdoor property of $\Pi_{\mathtt{com}}$. Let CH be the challenger for the trapdoor security game for ambiguous commitments.

We can construct an adversary $\mathcal{A}^{\mathsf{trap}}$ that interacts with $\mathcal{A}$ in the left and the right session according to both H_0 and H_1 for all messages except for the way the commitments are computed. For these messages the reduction acts as a proxy between $\mathcal{A}$ and CH in the left session. More formally, the reduction $\mathcal{A}^{\mathsf{trap}}$ proceeds as follows:

1. In the left session, on input the challenge $(i_1, q_{i_1}), \ldots, (i_k, q_{i_k})$, $i_j \in [k]$ and $q_{i_j} \in [c]$, compute $\{\mathsf{view}^{\mathtt{off}}_{i,q}\}_{i\in[n],q\in[c]}$ as in the honest prover procedure.
2. Let $I = \{i_1, \ldots, i_k\}$ and C_{i_j} be the set of all indices in $[c]/q_{i_j}$, for each $j \in [k]$. Send to the challenger CH messages $\{\mathsf{view}^{\mathtt{off}}_{i_1,l}, \ldots, \mathsf{view}^{\mathtt{off}}_{i_k,l}\}_{l\in C_{i_j}}$ obtaining $\{(\mathtt{com}_{(i_j,l)}, \mathtt{dec}_{(i_j,l)})\}_{i_j\in I, l\in C_{i_j}}$.
3. Compute π^1 following H_2 (and H_1) using $\{\mathtt{com}_{(i_j,l)}\}_{i_j\in I,l\in C_{i_j}}$.
4. Upon receiving x, w from the adversary $\mathcal{A}$ compute π^3 following H_2 (and H_1) and using $\{(\mathtt{com}_{(i_j,l)}, \mathtt{dec}_{(i_j,l)})\}_{i_j\in I,l\in C_{i_j}}$.
5. Finally, $\mathcal{A}^{\mathsf{shvzk}}$ wants to extract from the right session the messages committed in Π_{ComExt} using the corresponding extractor $\mathsf{Ext}_{\mathtt{com}}$ to feed this message in the distinguisher for H_1 and H_0. Therefore the reduction applies $\mathsf{Ext}_{\mathtt{com}}$ (w.l.o.g. we can assume that $\mathsf{Ext}_{\mathtt{com}}$ rewinds from the 3rd to the 2nd round since we are in the plain-model and Π_{ComExt} is 3-round). The only caveat is that during the rewinds also the left session is rewinded and $\mathcal{A}$ could choose a new theorem and witness x', w' for which it is expecting to obtain a new third round π^3 of the left session. This is not a problem since the MPC protocol $\Pi^{\mathtt{off},\mathtt{on}}$ is input independent so new views of the online phase can be generated w.r.t. x', w' and the same offline views can be reused (i.e., the same commitment openings).
6. The adversary runs the distinguisher for H_0 and H_1 (that exists by contradiction) on input the view of $\mathcal{A}$ and the committed message, and output whatever such distinguisher outputs.

□

Claim 2 *Let $\bar{p}$ be the probability that $\mathcal{A}$ changes the distribution of the committed messages between H_1 and H_2, then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

Proof. Suppose by contradiction that Claim 2 does not hold. Then it is possible to make a reduction that contradicts the t_p-privacy of $\Pi^{\mathtt{off},\mathtt{on}}$. Let CH be the challenger for t_p-privacy security game.

Then, we can construct an adversary $\mathcal{A}^{\mathsf{mpc}}$ that interacts with $\mathcal{A}$ in the left and the right session according to both H_2 and H_1 for all messages except for the way the online and offline views of $\Pi^{\mathtt{off},\mathtt{on}}$ are computed. For these messages, the reduction acts as a proxy between $\mathcal{A}$ and CH in the left session. More formally, the reduction $\mathcal{A}^{\mathsf{mpc}}$ proceeds as follows:

1. In the left session, on input the challenge $(i_1, q_{i_1}), \ldots, (i_k, q_{i_k})$, $i_j \in [k]$ and $q_{i_j} \in [c]$, compute π^1 following H_2 (and H_1).

2. Upon receiving x, w, send $I = \{i_1, \ldots, i_k\}$ and x, w to CH thus obtaining the views $\mathsf{view}^{\mathtt{off}}_{i_1}, \ldots, \mathsf{view}^{\mathtt{off}}_{i_k}, \mathsf{view}^{\mathtt{on}}_{i_1}, \ldots, \mathsf{view}^{\mathtt{on}}_{i_k}$.
3. Compute π^3 following H_2 (and H_1) and using the views $\mathsf{view}^{\mathtt{off}}_{i_1}, \ldots, \mathsf{view}^{\mathtt{off}}_{i_k}, \mathsf{view}^{\mathtt{on}}_{i_1}, \ldots, \mathsf{view}^{\mathtt{on}}_{i_k}$.
4. Finally, $\mathcal{A}^{\mathsf{mpc}}$ wants to extract from the right session the messages committed in Π_{ComExt} using the corresponding extractor $\mathsf{Ext}_{\mathtt{com}}$ o feed this message in the distinguisher for H_2 and H_1. Therefore the reduction applies $\mathsf{Ext}_{\mathtt{com}}$ (rewinding from the 3rd to the 2nd round). The only caveat is that during the rewinds also the left session is rewinded and $\mathcal{A}$ could choose a new theorem and witness x', w' for which it is expecting to obtain a new third round π^3 of the left session. This is not a problem since we can ask the challenger for a new set of views (that can be either simulated or generated honestly) due to the fact that t_p-privacy composes in parallel. Moreover, the reduction can keep fixed the first round while opening to the new offline views returned by the challenger due to the equivocality of the commitments.
   Following the arguments of the previous proofs, if the extraction probability changes (i.e., the probability of the adversary providing an accepting transcript) then we can already break the t_p-privacy property. If the extraction probability stays the same, the reduction can run the distinguisher for H_2 and H_1 (that exists by contradiction) on input the view of $\mathcal{A}$ and the extracted message, and return whatever the distinguisher outputs.

□

We recall that the commitment scheme $\Pi_{\mathtt{com}}$ used in Π_{AI} can be instantiated with any NI statistically binding scheme, which can be constructed from any one-to-one OWF. In addition, following [IKOS07], when we say that our protocols make black-box use of $\Pi^{\mathtt{off},\mathtt{on}}$, it simply means that they are invoking the "next-message function" of each party. Therefore, when $\Pi_{\mathtt{com}}$ is implemented using a black-box reduction to one-way functions, the protocol Π_{AI} only makes black-box use of one-way functions. More formally,

Corollary 1. *Assuming the existence of one-to-one one-way functions, there exists a 3-round public-coin delayed-input protocol satisfying adaptive-input soundness (with constant soundness error), and adaptive-input SHVZK, which makes black-box use of* 1-1 *OWFs. Moreover, let Π_{ComExt} be a 3-round extractable commitment scheme with a polynomial time extractor, that extracts with non-negligible probability, then there exists a 3-round public-coin delayed-input protocol that is non-malleable HVZK with respect to commitment for Π_{ComExt} against synchronizing adversaries that makes black-box use of the* 1-1 *OWFs.*

7 The Building Blocks of the 4-Round Black-Box Non-Malleable Commitment Scheme

In this section we define the main building blocks necessary to define our 4-round non-malleable commitment scheme.

7.1 Commitment from Verifiable Secret Sharing

We start by recalling some of the techniques introduced by Goyal et al. [GLOV12]. We show that these techniques can be used to build a Σ-commitment (Definition 4) that we denote by $\Pi = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ and formally describe it in Figure 3. The protocol makes use of the following primitives:

COMMON INPUTS: Both $\mathcal{S}^{\Sigma}$ and $\mathcal{R}^{\Sigma}$ get parameters t, n, k, where t, n are the parameters corresponding to the VSS $\Pi^{\mathsf{vss}} = (\Pi_{\mathsf{Share}}, \Pi_{\mathsf{Recon}})$, and $k \leq t$.
PRIVATE INPUT: At the beginning $\mathcal{S}^{\Sigma}$ gets a private message w.

Commitment procedure: $(\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma})$

Round 1. $\mathcal{S}^{\Sigma}$ proceeds as follows.

1. Run the sharing phase of Π^{vss} "in its head" on input w to generate the views view_j^{σ}, for each $j \in [n]$.
2. Compute $(\mathtt{com}_j^{\sigma}, \mathtt{dec}_j^{\sigma}) \xleftarrow{\$} \mathsf{Com}(\mathsf{view}_j^{\sigma})$ for $j \in [n]$.
3. Set
   - $\mathtt{dec}^{\sigma} \leftarrow \{\mathtt{dec}_j^{\sigma}, \mathsf{view}_j^{\sigma}\}_{j \in [n]}$
   - $\pi_1^{\sigma} \leftarrow (\mathtt{com}_1^{\sigma}, \ldots, \mathtt{com}_n^{\sigma})$
4. Send π_1^{σ} to $\mathcal{R}^{\Sigma}$.

Round 2. $\mathcal{R}^{\Sigma}$ executes the following steps.

1. Choose a random subset $I \leftarrow \{i_1, \ldots, i_k\} \subset [n]$ and send it to $\mathcal{S}^{\Sigma}$.

Round 3. $\mathcal{S}^{\Sigma}$ computes the following steps:

1. Define and send $\pi_3^{\sigma} = \{\mathsf{view}_j^{\sigma}, \mathtt{dec}_j^{\sigma}\}_{j \in I}$ to $\mathcal{R}^{\Sigma}$.
2. Set $\mathtt{com}_{\sigma} = (\pi_1^{\sigma}, \pi_2^{\sigma}, \pi_3^{\sigma})$

Verification step. $\mathcal{R}^{\Sigma}$ accepts the commitment if and only if:

1. $\mathsf{Dec}(\mathtt{com}_j^{\sigma}, \mathsf{view}_j^{\sigma}, \mathtt{dec}_j^{\sigma}) = 1$ and the output of $\mathtt{P}_j$ in Π^{vss} is not $\perp$, for each $j \in I$.
2. The views $\mathsf{view}_{i_1}^{\sigma}, \ldots, \mathsf{view}_{i_k}^{\sigma}$ are consistent.

Decommitment procedure: $\mathsf{Dec}^{\Sigma}(\mathtt{com}_{\sigma}, w, \mathtt{dec}_{\sigma})$

1. Parse $\mathtt{dec}^{\sigma}$ as $\{\mathtt{dec}_j^{\sigma}, \mathsf{view}_j^{\sigma}, w_j\}_{j \in [n]}$.
2. Use $\{\mathsf{view}_j^{\sigma}\}_{j \in [n]}$ as the inputs of Π_{Recon} thus obtaining w.
3. Check that for all $j \in [n]$ it holds that $\mathsf{Dec}(\mathtt{com}_j^{\sigma}, \mathsf{view}_j^{\sigma}, \mathtt{dec}_j^{\sigma}) = 1$.

If the above conditions hold, $\mathcal{R}^{\Sigma}$ outputs w, else it returns $\perp$.

Fig. 3: $\Pi = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$

- An $(n+1, t)$-VSS protocol $\Pi^{\mathsf{vss}} = (\Pi_{\mathsf{Share}}, \Pi_{\mathsf{Recon}})$ as defined in Definition 14. Concretely, the protocol uses a VSS scheme with a deterministic reconstruction procedure, like the $(n+1, \lfloor n/4 \rfloor)$-VSS scheme described by Gennaro et al. [GIKR01]
- A statistically binding commitment scheme $\Pi^{\mathtt{com}} = (\mathsf{Com}, \mathsf{Dec})$.

The protocol works as follows. To commit to a message w, the sender $\mathcal{S}^{\Sigma}$ runs "in its head" the protocol Π_{Share}, which implements the sharing phase of Π^{vss}, with input w. Then the sender commits to the views view_j (obtained by the execution of Π_{Share}) of each $\mathtt{P}_j$ separately using a statistical binding commitment scheme $\Pi^{\mathtt{com}}$. The receiver, upon receiving these commitments, samples a random set $I \subset [n]$, with $|I| \leq t$, and sends it to the sender. Finally, the sender replies by decommitting the views corresponding to the challenge I. This concludes the commit phase.

We prove now the following theorem that we shall use in the next sections.

Theorem 3. *Let Π^{vss} be a perfectly secure $(n+1, t)$-VSS protocol satisfying Definition 14, with $t = k$, $t < \frac{1}{4}n$, and let $\Pi^{\mathtt{com}}$ be a statistically binding commitment scheme, then $\Pi = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ (see Figure 3) is a Σ-commitment with $\binom{n}{k}$-special binding.*

Proof. CORRECTNESS. It follows by inspections.
HONEST RECEIVER HIDING. To prove this property we proceed trough a series of hybrids.

H_0**:** This hybrid takes as input the challenge π_2^{σ} and w, and computes the first and the third round of Π using the honest procedure of $\mathcal{S}^{\Sigma}$ w.r.t. the message w.

H_1: This hybrid, on input the challenge π_2^σ and w, proceeds as follows.
Let I be the set of indices contained in π_2^σ:
- For each $i_j \notin I$, compute $(\mathtt{com}_{i_j}, \mathtt{dec}_{i_j}) \leftarrow \mathsf{Com}(0^\ell)$;
- For each $i_j \in I$, $(\mathtt{com}_{i_j}, \mathtt{dec}_{i_j}) \leftarrow \mathsf{Com}(\mathsf{view}_{i_j}^\sigma)$.

The rest of the hybrid is executed as H_0, in particular the views $\{\mathsf{view}_{i_j}^\sigma\}_{i_j \in I}$ are computed as before. This hybrid is indistinguishable from the previous one due to the hiding property of the commitment scheme.

H_2: This hybrid is defined as the previous one except that the views $\{\mathsf{view}_{i_j}^\sigma\}_{i_j \in I}$ of parties in I are computed using the simulator of Π^{vss}. This hybrid corresponds to the simulator Sim^σ for Π and it is indistinguishable from H_1 due to the secrecy property of Π^{vss}.

$\binom{n}{k}$-Special Binding. It is easy to see that having a third round of the protocol for all the possible challenges it allows the extraction of the committed in ppt.

7.2 Commit-and-Prove

In this section we construct a 3-round public-coin commit-and-prove protocol $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ that allows proving the knowledge of a committed value w such that $\mathsf{Rel}(x, w) = 1$, for some statement x. The protocol $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ is fully described in Figure 4. It makes use of the following tools:

- The Σ-commitment $\Sigma = ((\mathcal{S}^\Sigma, \mathcal{R}^\Sigma), \mathsf{Dec}^\Sigma)$ defined in Figure 3, Section 7.1 with 28-special-binding (from Theorem 3 this can be obtained by setting $n = 8, k = 2$).
- The adaptive-input SHVZK $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ with adaptive-input soundness (and negligible soundness error) for the $\mathcal{NP}$-relation

$$\mathsf{Rel}_{\mathsf{AI}} = \{(x, a, \alpha, \{\mathsf{views}_i'\}_{i\in[\lambda]}), (\{r_i, \mathsf{views}_i\}_{i\in[\lambda]}) : \mathsf{views}_i' \subseteq \mathsf{views}_i,$$
$$k = |\mathsf{views}_i'| < |\mathsf{views}_i| = n \mid \forall i \in [\lambda]\ w_i = \mathsf{Recon}(\mathsf{views}_i) \ \wedge\ \mathsf{Rel}(x, w_i) = 1 \ \wedge\ a = w_i + r_i\alpha\},$$

  where Recon is the reconstruction phase of an information-theoretic $(9, 2)$-VSS protocol Π^{vss}. We recall that to run Π_{AI} the prover needs statement and witness only in the third round. We note that given that Π^{vss} is information-theoretic, then Π_{AI} still makes black-box use of the underlying cryptographic primitives.

Theorem 4. *Let Π_{ComExt} be a 3-round extractable commitment scheme, let $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ be a 3-round public-coin, delayed-input complete, adaptive-input NMZKC for Π_{ComExt} adaptive-input soundness (with negligible soundness error) for the $\mathcal{NP}$-relation $\mathsf{Rel}_{\mathsf{AI}}$, and $\Sigma = ((\mathcal{S}^\Sigma, \mathcal{R}^\Sigma), \mathsf{Dec}^\Sigma)$ (as defined in Figure 3) be a Σ-commitment with $n = 8, k = 2$, then $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ is a 3-round public-coin adaptive-theorem NMZKC for Π_{ComExt}, and it is a commit-and-prove protocol for the $\mathcal{NP}$-relation Rel.*

Proof. Commit-and-prove. To prove this theorem we need to prove two facts. The first is that if an adversarial sender provides an accepting transcript with some non-negligible probability p, then there exists a valid opening for that transcript. The second property we need to prove is that there exists a ppt extractor that returns a message $m \neq \perp$, that corresponds to the only massage to which the commitment can be opened.

PUBLIC INPUT AND PARAMETERS: Parameters k, n, t of Σ, with $k = t$, where $n = 8$ and $t = k = 2$. $\mathcal{P}_{\mathsf{CP}}$ and $\mathcal{V}_{\mathsf{CP}}$ gets x in the third round.
PRIVATE INPUT: At the beginning $\mathcal{P}_{\mathsf{CP}}$ gets w.

Round 1 $\mathcal{P}_{\mathsf{CP}}$ executes the following steps:
1. Sample $r \leftarrow \mathbb{F}$.
2. For $i \in [\lambda]$, do the following:
   2.1. Execute $\mathcal{S}^{\Sigma}$ on input $(1^{\lambda}, w||r)$, obtaining $\pi^{\sigma}_{i,1} = \{\mathtt{com}^{\sigma}_{i,j}\}_{j\in[n]}$, $\{\mathtt{dec}^{\sigma}_{i,j}\}_{j\in[n]}$ and $\{\mathsf{view}^{\sigma}_{i,j}\}_{j\in[n]}$.
   2.2. Run $\mathcal{P}_{\mathsf{AI}}$ thus obtaining $\pi_{i,1}$.
3. Define and send $\pi_1 = \{\pi_{i,1}, \pi^{\sigma}_{i,1}\}_{i\in[\lambda]}$ to $\mathcal{V}_{\mathsf{CP}}$.

Round 2 $\mathcal{V}_{\mathsf{CP}}$ computes the following steps:
1. For $i \in [\lambda]$, run $\mathcal{V}_{\mathsf{AI}}$ thus obtaining $\pi_{i,2}$ and run $\mathcal{R}^{\Sigma}$ thus obtaining $\pi^{\sigma}_{i,2}$
2. Sample $\alpha \leftarrow \mathbb{F}$.
3. Set $\pi_2 = (\{\pi_{i,2}\}_{i\in[\lambda]}, \{\pi^{\sigma}_{i,2}\}_{i\in[\lambda]}, \alpha)$ and send it to $\mathcal{P}_{\mathsf{CP}}$.

Round 3 $\mathcal{P}_{\mathsf{CP}}$ performs the following steps:
1. Compute $a = w + r\alpha$ and, for each $i \in [\lambda]$, do as follows.
   1.1. Compute the 3rd message $\pi^{\sigma}_{i,3}$ of Σ executing $\mathcal{S}^{\Sigma}$ on input $\pi^{\sigma}_{i,2}$ (note that $\pi^{\sigma}_{i,3} = \{\mathtt{dec}^{\sigma}_{i,j}, \mathsf{view}^{\sigma}_{i,j}\}_{j\in\pi^{\sigma}_{i,2}}$).
   1.2. Define $\mathsf{views}'_i \leftarrow \{\mathsf{view}^{\sigma}_{i,j}\}_{j\in\pi^{\sigma}_{i,2}}$ and $\mathsf{views}_i \leftarrow \{\mathsf{view}^{\sigma}_{i,j}\}_{j\in[n]}$.
2. For each $i \in [\lambda]$ Run $\mathcal{P}_{\mathsf{AI}}$ on input the pair statement-witness $((x, a, \alpha, \{\mathsf{views}'_c\}_{i\in\lambda}), \{\mathsf{views}_c\}_{c\in\lambda})$ and $\pi_{i,2}$, thus obtaining the third round $\pi_{i,3}$.
3. Set $\pi_3 = (\{\pi^{\sigma}_{i,3}\}_{i\in[\lambda]}, \{\pi_{i,3}\}_{i\in[\lambda]}, a)$ and send π_3 to $\mathcal{V}_{\mathsf{CP}}$.

Verification step. On input x, outputs 1 if and only if, for each $i \in [\lambda]$, the following holds:
1. $\mathcal{R}^{\Sigma}$ accepts the commitment $(\pi^{\sigma}_{i,1}, \pi^{\sigma}_{i,2}, \pi^{\sigma}_{i,3})$.
2. $\mathcal{V}_{\mathsf{AI}}$ accepts the proof $(\pi_{i,1}, \pi_{i,2}, \pi_{i,3})$ for the statement $(x, a, \alpha, \{\mathsf{views}'_c\}_{i\in\lambda})$, where views'_c is defined as before.

Decommitment procedure: On input an accepting transcript of the protocol, and on input all the decommitment information for the λ commitments generated via Σ, return m, if and only if the majority of the Σ-commitments are commitments of $(m\| \cdot)$.

Fig. 4: $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$

To prove the first claim, we need to argue that if a receiver accepts a proof, then the majority of the Σ-commitments are well formed (this is due to how the opening procedure of Π_{CP} is defined). We prove this as follows. Let us consider a single execution of a Σ-commitment (parametrized with $n = 8$ and $k = 2$) against a corrupted sender. The best strategy for the adversarial prover to provide an accepting transcript for a single execution of a Σ-commitment, while at the same time not being detected, is to compute in an ill-formed way 2 views (note that the two views might be consistent with each other, even if they are not consistent with the remaining 6 views). We denote this strategy with best, and argue that this is the best strategy that an adversary could implement in order to not get caught, while still computing an accepting transcript for a Σ-commitment that does not have a valid opening. The probability of an adversary that completes an accepting transcript for a Σ-commitment with some probability p_σ of not getting caught using the strategy best is at most $p_\sigma(\frac{\binom{n-2}{2}+1}{\binom{n}{2}})$. Note that $\frac{\binom{n-2}{2}+1}{\binom{n}{2}} \geq \frac{\binom{n-i}{2}+\binom{i}{2}}{\binom{n}{2}}$ for all $n > 4, 2 \leq y \leq n-2$, where $\binom{n-i}{2} + \binom{i}{2}$ denotes the number of accepting transcripts for a Σ-commitment in the case where the adversary decides to generate a set A of size $(n-i)$ of views consistent with each other, and a set B of size i of views (consistent with each other), where each view A is inconsistent with a view in B.[12].

Note also that a strategy where the adversary computes only one view in an ill-formed way does not represent an attack since, due to the commitment property of the VSS scheme that underlines

[12] Observe that this holds since a verifier would reject any transcript in which one view from A and one view from B is opened

the Σ-commitment, it is still possible to reconstruct the message correctly even if one (or two views) are ill-formed. Given that best is the best strategy for the adversary to compute a bad execution of a Σ commitment while not getting caught (i.e., a Σ-commitment that commits to a value $\perp$), then we can claim that the probability of the adversary getting caught, when $n = 8$ and $k = 2$, is at least $(1 - 4/7)p_\sigma > 1/3p_\sigma$ (where p_σ is the probability of an adversary of providing an accepting transcript for one execution of a Σ-commitment).

Going back to the PoK proof Π_{CP}. As we mentioned, the first thing we want to prove is that if the adversarial sender provides an accepting transcript for Π_{CP} with non-negligible probability p, then the majority of the Σ-commitments are computed correctly.

Suppose by contradiction that more than $\lambda/3$ commitments are ill-formed (i.e., at least two views are inconsistent with each other). Without loss of generality, we can assume that the adversary follows the strategy best to compute the ill-formed Σ commitments. In this case, the probability that the receiver accepts the transcript of Π_{CP} is bounded by $(1 - p_\sigma/3)^{\lambda/3} = \nu(\lambda)$, which is a contradiction since we have assumed that the receiver accepts with non-negligible probability p. Hence, the number of ill-formed transcripts of Σ-commitment must be less than $\lambda/3$, as such we can claim that the majority of the Σ-commitment are well-formed. In the rest of the proof we will denote the set of all the well-formed Σ-commitments with S. We can now go to the second part of the proof and show that all the well-formed Σ-commitment are commitments of the same value and that we can extract this value in PPT.

We first recall that due to the commitment property of the VSS we can claim that there is no set of honestly generated views $\{\mathsf{view}_i, \mathsf{view}'_i\}_{i_j \notin \{\alpha,\beta\}}$ such that $\mathsf{Recon}(\mathsf{view}_1, \ldots, \mathsf{view}_n) = m$ and $\mathsf{Recon}(\mathsf{view}'_1, \ldots, \mathsf{view}'_n) = m'$ where $m \neq m'$ and $\mathsf{view}_\alpha = \mathsf{view}'_\alpha$ and $\mathsf{view}_\beta = \mathsf{view}'_\beta$. This, together with the soundness of Π_{AI} guarantees that for each Σ-commitment transcript in S, the committed value $w_i||r_i$ is such that $w_i\alpha + r_i = a$ with overwhelming probability over α and $(x, w_i) \in \mathsf{Rel}$. By Schwartz-Zippel lemma, this is possible only if there exists (w, r) such that $w_i = w$ and $r_i = r$ for all $\Sigma_i \in S$. Hence, the majority of the Σ-commitment are well formed and contain the same value $w||r$.

It remains to argue that we can extract such w in PPT against an adversary that provides an accepting transcript with non-negligible probability p.

By definition, the Σ-commitment we use is 28-special binding. This means that there exists a polynomial time algorithm, that on input all the possible accepting transcripts for a Σ-commitment, returns the committed value. Our extractor works as follows. It acts as the honest receiver, and upon receiving an accepting transcript, it rewinds the adversary exactly 28 times. If the extractor manages to get all the possible accepting transcripts for one execution of a Σ-commitment, then the extractor can run the special-binding extractor and returns what the extractor for special-binding returns, else it returns $\perp$.

This extractor is clearly PPT and it extracts the correct message. Moreover, given that it is possible to extract all the possible accepting transcripts from a Σ-commitment only if it is well-formed, from the argument above we can then claim that the extracted value is the correct one. The only thing it remains to argue is that the extractor returns a value different from $\perp$ with non-negligible probability. The probability that the adversary provides 28-accepting transcripts in each rewind performed by our simulator is p^{28}. The probability that, for a random execution of a Σ-commitment, every randomly sampled challenge contains a new challenge for the i-th execution of the Σ-commitment is at leaset $1/n = 1/8$. Hence, the probability that the extractor manages to

collect all the possible accepting transcript for a random execution of a Σ-commitment is at least $(p/9)^{28}$, which represents a non-negligible value since p, by assumption, is non-negligible.

ADAPTIVE-THEOREM NMZKC.

Let $(\mathsf{Sim}^0_{\mathsf{AI}}, \mathsf{Sim}^1_{\mathsf{AI}})$ be the NMZKC of of Π_{AI}, and let Sim^σ be the simulator for the Σ-commitment. Our simulator $\mathsf{Sim}_{\mathsf{CP}} = (\mathsf{Sim}^0_{\mathsf{CP}}, \mathsf{Sim}^1_{\mathsf{CP}})$ works as follows.

$\mathsf{Sim}^0_{\mathsf{CP}}$. On input challenge (α, π_2) and the length of the theorem 1^m, parse π_2 as $\{\pi^\sigma_{i,2}\}_{i\in[\lambda]}, \{\pi_{i,2}\}_{i\in[\lambda]}$ and compute the following steps:

1. For all $i \in [\lambda]$:
   1.1. Run Sim^σ on input $\{\pi^\sigma_{i,2}\}_{i\in[\lambda]}$ to generate $\pi^\sigma_{i,1}, \pi^\sigma_{i,3}$.
   1.2. Run $\mathsf{Sim}^0_{\mathsf{AI}}$ on input 1^m and $\{\pi_{i,2}\}_{i\in[\lambda]}$ to generate $(\pi_{i,1}, \mathsf{aux}_i)$.
2. Output $\{\pi_{i,1}, \pi^\sigma_{i,1}\}_{i\in[\lambda]}$.

$\mathsf{Sim}^1_{\mathsf{CP}}$. On input the theorem x do the following steps.

1. Choose a at random.
2. For all $i \in [\lambda]$, parse $\pi^\sigma_{i,3}$ as $\{\mathsf{dec}^\sigma_{i,j}, \mathsf{view}^\sigma_{i,j}\}_{j\in\pi^\sigma_{i,2}})$ and define $\mathsf{views}'_i \leftarrow \{\mathsf{view}^\sigma_{i,j}\}_{j\in\pi^\sigma_{i,2}}$
3. For all $i \in [\lambda]$ run $\mathsf{Sim}^1_{\mathsf{AI}}$ on input $X = ((x, a, \alpha, \{\mathsf{views}'_i\}_{i\in[\lambda]})$ and aux_i thus obtaining the third round $\pi_{i,3}$.
4. Set $\pi_3 \leftarrow (\{\pi^\sigma_{i,3}\}_{i\in[\lambda]}, \{\pi_{i,3}\}_{i\in[\lambda]}, a)$ and output π_3.

We will now briefly argue that our simulator satisfies the notion of adaptive-theorem non-malleable HVZK with respect to commitment.

H_0: This hybrid generates the transcript in the left session running the honest prover. In the right session the hybrid interacts as honest receiver with the adversary $\mathcal{A}$.

H_1: This hybrid is identical to the previous one except that, in the left session, the transcript of Π_{AI} is generated using the NMZKC simulator $\mathsf{Sim}_{\mathsf{AI}} = (\mathsf{Sim}^0_{\mathsf{AI}}, \mathsf{Sim}^1_{\mathsf{AI}})$.
The indistinguishability between the two hybrid and the fact that the distribution of the messages committed on the right session using Π_{ComExt} comes immediately from the NMZKC for Π_{ComExt}.

H_2: This hybrid is identical to the previous one except that it runs Sim^σ to generate the transcript of Σ.
The two hybrids are indistinguishable due to the (parallel composable) honest receiver hiding of Σ. Moreover, we can prove that the distribution of the messages committed via Π_{ComExt} does not change as this would contradict the receiver hiding of Σ. Note that a reduction to the hiding of Σ can be done since the rewinds made to extract the message committed in Π_{ComExt} do not rewind the challenger since the challenge and the witness that eventually the challenger needs to commit to are fixed at the on-set of the experiment.

H_3: The hybrid is identical to H_2, but it changes the way the value a, sent in the third round, is computed. In particular, a is chosen at random, instead of following the honest prover procedure. These two hybrids are statistically indistinguishable due to the fact that a hides the witness w information theoretically.

□

Corollary 2. *Let Π_{ComExt} be a 3-round extractable commitment scheme and Rel be an NP-relation. Assuming the existence of one-to-one one-way functions, there exists a 3-round public-coin adaptive-theorem NMZKC for Π_{ComExt}, a commit-and-prove protocol for the $\mathcal{NP}$-relation Rel that makes black-box use of the 1-1 OWFs.*

PUBLIC PARAMETERS: 1^λ, ℓ, tags $\mathsf{tg}_1, \ldots, \mathsf{tg}_\ell$ and a large prime q s.t. $q > 2^{\mathsf{tg}_i}$ for all i. A default second round message π_2^σ for Σ (i.e., $\pi_2^\sigma = \{1, 2, \ldots, k\}$).
PRIVATE INPUT: $\mathcal{S}_{\mathsf{wnmc}}$ gets $m \in \mathbb{F}_q$.

Round 1 $\mathcal{S}_{\mathsf{wnmc}}$ computes the following steps:

1. Pick at random $r_1, \ldots, r_\ell$ and perform λ^2 executions of $\mathcal{S}^\Sigma$ on input $(1^\lambda, m||r_1||, \ldots, ||r_\ell)$, thus obtaining $\pi_1^\sigma = \{\pi_{1,i}^\sigma\}_{i\in[\lambda^2]}$ and $\mathtt{dec}^\sigma = \{\mathtt{dec}_i^\sigma\}_{i\in[\lambda^2]}$. Send π_σ^1 to $\mathcal{R}_{\mathsf{wnmc}}$.

Round 2 $\mathcal{R}_{\mathsf{wnmc}}$ computes the following steps:

1. Pick at random challenge vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_\ell)$, where $\alpha_i \in [2^{\mathsf{tg}_i}] \subset \mathbb{F}_q$.
2. Send $\vec{\alpha}$ to $\mathcal{S}_{\mathsf{wnmc}}$.

Round 3 $\mathcal{S}_{\mathsf{wnmc}}$ computes the following steps:

1. Compute the third message π_3^σ of Σ executing $\mathcal{S}^\Sigma$ on input π_2^σ.
2. For all $i \in [\ell]$, compute $a_i \leftarrow r_i\alpha_i + m$, set $\vec{a} = (a_1, \ldots, a_\ell)$.
3. Send $(\pi_3^\sigma, \vec{a})$ to $\mathcal{R}_{\mathsf{wnmc}}$.

Decommitment procedure: On input an accepting transcript of the protocol, and on input all the decommitment information for the λ commitments generated via Σ, return m, if and only if the majority of the Σ-commitments are commitments of $(m||r_1||, \ldots, ||r_\ell)$, and for all $i \in [\ell]$ it holds that $a_i = r_i\alpha_i + m$.

Fig. 5: $\Pi_{\mathsf{wnmc}} = (\mathcal{S}_{\mathsf{wnmc}}, \mathcal{R}_{\mathsf{wnmc}})$

Remark 1. To simplify the exposition of our non-malleable commitment scheme that internally uses the commit-and-prove protocol we have just described, we will consider the messages of Π_{CP} as divided into two parts: the messages related to the proof phase, and the messages related to the commitment phase.

7.3 The 4-Round Non-Malleable Commitment Scheme of [GRRV14a]

The 4-round non-malleable commitment of Goyal et al. [GRRV14a] is composed of two parts: the first one is a special public-coin Π_{wnmc} commitment scheme, that enjoys a weak form of non-malleability. Loosely speaking, Π_{wnmc} is non-malleable as long as the MiM, acting as a sender, is committing to a well-formed commitment. The second part is a zero-knowledge PoK that ensures that Π_{wnmc} is computed correctly. In Figure 5, we recall the protocol Π_{wnmc}. This uses as an underlying building block a non-interactive commitment that is statistically binding. We replace this commitment with our interactive Σ-commitment Σ where the challenge is a default value (i.e., this trivially makes the Σ-commitment non-interactive). Finally, we prove that, after this modification, Π_{wnmc} remains hiding.

Lemma 3. *Let Σ be the Σ-commitment described in Figure 3, then $\Pi_{\mathsf{wnmc}} = (\mathcal{S}_{\mathsf{wnmc}}, \mathcal{R}_{\mathsf{wnmc}})$ described in Figure 5 enjoys the hiding property.*

This follows from Theorem 3 $a_1, \ldots, a_\ell$ information theoretically hide the committed message.

8 Our 4-Round Black-Box Non-Malleable Commitment Scheme

Our 4-round non-malleable commitment $\Pi_{\mathsf{nmc}} = ((\mathcal{S}_{\mathsf{nmc}}, \mathcal{R}_{\mathsf{nmc}}), \mathsf{Dec}_{\mathsf{nmc}})$ makes use of the following tools.

- An ambiguous commitment scheme $\Pi_{\mathtt{com}} = (\mathsf{Com}, \mathsf{Dec}, \mathsf{Com}^{\mathsf{eq}}, \mathsf{Eq})$ as described in Section 3.6.

COMMON INPUTS: 1^λ, Parameters. k, n, t of Σ, with $k = t$, where $n = 8$ and $t = k = 2$, $\ell = \lambda$, the tags $\mathtt{tg}_1, \ldots, \mathtt{tg}_\ell$ and a large prime q s.t. $q > 2^{\mathtt{tg}_i}$ for all $i \in [\ell]$. m, as described above.
PRIVATE INPUT: At the beginning $\mathcal{S}_{\mathsf{nmc}}$ gets $m \in \mathbb{F}_q$.

Round 1. $\mathcal{R}_{\mathsf{nmc}}$ picks two random strings $\hat{s}_0, \hat{s}_1$, and a random bit b and runs $\mathcal{P}_{\mathsf{comWI}}$ on input $(1^\lambda, \hat{s}_b, b)$ thus obtaining π_1^{comWI} and sends it to $\mathcal{S}_{\mathsf{nmc}}$.

Round 2. $\mathcal{S}_{\mathsf{nmc}}$ executes the following steps:

1. Compute the 1st round of Π_{wnmc}: Pick ℓ random strings $r_1, \ldots, r_\ell$ and for $i \in [\lambda]$, do the following:
   1.1. Execute $\mathcal{S}^\Sigma$ on input $(1^\lambda, (m, r_1, \ldots, r_\ell))$, obtaining $\pi^\sigma_{i,1} = \{\mathtt{com}^\sigma_{i,j}\}_{j\in[n]}$, $\{\mathtt{dec}^\sigma_{i,j}\}_{j\in[n]}$ and $\{\mathsf{view}^\sigma_{i,j}\}_{j\in[n]}$.
   1.2. Run $\mathcal{P}_{\mathsf{AI}}$ on input 1^λ thus obtaining $\pi_{i,1}$.
2. Define $\pi_1 = \{\pi_{i,1}\}_{i\in[\lambda]}$ and $\pi_1^\sigma = \{\pi^\sigma_{i,1}\}_{i\in[\lambda]}$.
3. For each $i \in [\lambda]$ sample $s_i^0 \xleftarrow{\$} \{0,1\}^{\lambda^2}$, set $s_i^1 \leftarrow \pi_{i,1} \oplus s_i^0$, pick $R_i^b \xleftarrow{\$} \{0,1\}^\lambda$ and compute $(\mathtt{com}_i^b, \mathtt{dec}_i^b) \xleftarrow{\$} \mathsf{Com}(s_b; R_i^b)$ for each $b \in \{0,1\}$.
4. Sample a random string $\beta_0 \xleftarrow{\$} \{0,1\}^\lambda$.
5. For each $i \in [\lambda]$ sample $\overline{s}_i^0 \xleftarrow{\$} \{0,1\}^{\lambda^2}$, set $\overline{s}_i^1 \leftarrow \beta_0 \oplus \overline{s}_i^0$, pick $\overline{R}_i^b \xleftarrow{\$} \{0,1\}^\lambda$ and compute $(\overline{\mathtt{com}}_i^b, \overline{\mathtt{dec}}_i^b) \xleftarrow{\$} \mathsf{Com}(\overline{s}_b; \overline{R}_i^b)$ for each $b \in \{0,1\}$.
6. Run the simulator for Π_{tr} twice: Pick $\gamma_0 \xleftarrow{\$} \{0,1\}^m$, and run $\mathsf{Sim}_{\mathsf{tr}}$ on input $(\gamma_0, \gamma_0, \gamma_0^2)$ thus obtaining $(\mathsf{aux}, \pi_1^{\mathsf{tr}})$. Pick $\overline{\gamma}_0 \xleftarrow{\$} \{0,1\}^m$, and run $\mathsf{Sim}_{\mathsf{tr}}$ on input $(\overline{\gamma}_0, \overline{\gamma}_0, \overline{\gamma}_0^2)$ thus obtaining $(\overline{\mathsf{aux}}, \overline{\pi}_1^{\mathsf{tr}})$.
7. Compute the second round π_2^{comWI} of $\mathcal{V}_{\mathsf{comWI}}$.
8. Send $(\pi_1^\sigma, \pi_1^{\mathsf{tr}}, \overline{\pi}_1^{\mathsf{tr}}, \pi_2^{\mathsf{comWI}}, \{\mathtt{com}_i^b\}_{b\in\{0,1\}, i\in[\lambda]})$ to $\mathcal{S}_{\mathsf{nmc}}$.

Round 3 $\mathcal{R}_{\mathsf{nmc}}$ executes the following steps:

1. Compute the 2nd round of Π_{wnmc}: Pick a random challenge vector $\vec{\alpha} = (\alpha, \ldots, \alpha_\ell)$, where $\alpha_i \in [2^{\mathtt{tg}_i}] \subset \mathbb{F}_q$.
2. Run the third round π_3^{comWI} of $\mathcal{P}_{\mathsf{comWI}}$ on input $(\pi_2^{\mathsf{comWI}}, \hat{s}_1)$.
3. Run $\mathcal{V}_{\mathsf{AI}}$ λ times thus obtaining $\pi_2 = \{\pi_{i,2}\}_{i\in[\lambda]}$.
4. Pick $\beta_1 \xleftarrow{\$} \{0,1\}^m$, $\gamma_1 \xleftarrow{\$} \{0,1\}^m$, $\overline{\gamma}_1 \xleftarrow{\$} \{0,1\}^m$ and send $(\vec{\alpha}, \pi_3^{\mathsf{comWI}}, \hat{s}_0, \hat{s}_1, \pi_2, \alpha, \beta_1, \gamma_1, \overline{\gamma}_1)$ to $\mathcal{S}_{\mathsf{nmc}}$.

Round 4. $\mathcal{S}_{\mathsf{nmc}}$ computes the following steps:

1. If $\mathcal{V}_{\mathsf{comWI}}$ accepts the proof, $(\pi_1^{\mathsf{comWI}}, \pi_2^{\mathsf{comWI}}, \pi_3^{\mathsf{comWI}}, \hat{s}_0, \hat{s}_1)$ continue, else abort.
2. Set $\pi_2^\sigma = \{\pi^\sigma_{i,2}\}_{i\in[\lambda]} \leftarrow \beta_0 \oplus \beta_1$.
3. Compute the 3rd round of Π_{wnmc}: For all $i \in [\ell]$, compute $a_i = r_i\alpha_i + m$.
4. Define $x = \{a_i, \alpha_i\}_{i\in[\ell]}$ and set $w = (m, r_1, \ldots, r_\ell)$.
5. For each $i \in [\lambda]$ compute the 3rd message $\pi^\sigma_{i,3}$ of Σ executing $\mathcal{S}^\Sigma$ on input $\pi^\sigma_{i,2}$ (note that $\pi^\sigma_{i,3} = \{\mathtt{dec}^\sigma_{i,j}, \mathsf{view}^\sigma_{i,j}\}_{j\in\pi^\sigma_{i,2}}$), and define $\mathsf{views}'_i \leftarrow \{\mathsf{view}^\sigma_{i,j}\}_{j\in\pi^\sigma_{i,2}}$ and $\mathsf{views}_i \leftarrow \{\mathsf{view}^\sigma_{i,j}\}_{j\in[n]}$.
6. For each $i \in [\lambda]$ Run $\mathcal{P}_{\mathsf{AI}}$ on input the pair statement-witness $((x, \{\mathsf{views}'_u\}_{u\in\lambda}), \{\mathsf{views}_u\}_{u\in\lambda})$ and $\pi_{i,2}$, thus obtaining the third round $\pi_{i,3}$.
7. Set $\pi_3^\sigma \leftarrow \{\pi^\sigma_{i,3}\}$ and $\pi_3 \leftarrow \{\pi_{i,3}\}_{i\in[\lambda]}$.
8. Run $\mathsf{Sim}_{\mathsf{tr}}$ on input $(\mathsf{aux}, \hat{s}_0, \hat{s}_1)$ thus obtaining π_3^{tr}. Run $\mathsf{Sim}_{\mathsf{tr}}$ on input $(\overline{\mathsf{aux}}, \hat{s}_0, \hat{s}_1)$ thus obtaining $\overline{\pi}_3^{\mathsf{tr}}$
9. Set $c \leftarrow \gamma_0 \oplus \gamma_1$ and let c_i be the i-th bit of c with $i \in [\lambda]$. Analogously, $\overline{c} \leftarrow \overline{\gamma}_0 \oplus \overline{\gamma}_1$ and let $\overline{c}_i$ be the i-th bit of $\overline{c}$ with $i \in [\lambda]$.
10. Send $(\pi_2^{\mathsf{tr}} = (\gamma_0^2, \gamma_0, \gamma_0), \overline{\pi}_2^{\mathsf{tr}} = (\overline{\gamma}_0^2, \overline{\gamma}_0, \overline{\gamma}_0), \pi_3^{\mathsf{tr}}, \pi_3^\sigma, \pi_3, \{s_i^{c_i}, \mathtt{dec}_i^{c_i}\}_{i\in[\ell]}, \{s_i^{1-c_i}, R_i^{1-c_i}\}_{i\in[\ell]}, \{\overline{s}_i^{\overline{c}_i}, \overline{\mathtt{dec}}_i^{\overline{c}_i}\}_{i\in[\ell]}, \{\overline{s}_i^{1-\overline{c}_i}, \overline{R}_i^{1-\overline{c}_i}\}_{i\in[\ell]})$ to $\mathcal{R}_{\mathsf{nmc}}$.

Verification step. Set $c \leftarrow \gamma_0 \oplus \gamma_1$ and $\pi_2^\sigma = \{\pi^\sigma_{i,2}\}_{i\in[\lambda]} = \beta_0 \oplus \beta_1$ and accept the commitment if and only if:

1. $\mathcal{V}_{\mathsf{tr}}$ accepts the proofs for $(\pi_1^{\mathsf{tr}}, (\gamma_0, \gamma_0, \gamma_0^2), \pi_3^{\mathsf{tr}})$ with respect to the instance $(\hat{s}_0, \hat{s}_1)$
2. $\mathcal{V}_{\mathsf{tr}}$ accepts the proofs for $(\overline{\pi}_1^{\mathsf{tr}}, (\overline{\gamma}_0, \overline{\gamma}_0, \overline{\gamma}_0^2), \overline{\pi}_3^{\mathsf{tr}})$ with respect to the instance $(\hat{s}_0, \hat{s}_1)$
3. For each $i \in [\lambda]$
   - $\mathsf{Dec}(\mathtt{com}_i^{c_i}, s_i^{c_i}, \mathtt{dec}_i^{c_i}) = 1$, $\mathsf{Com}(s_i^{1-c_i}; R_1^{1-c_i}) = \mathtt{com}_i^{1-c_i}$ and $\pi_{i,1} = s_i^0 \oplus s_i^1$.
   - $\mathsf{Dec}(\overline{\mathtt{com}}_i^{\overline{c}_i}, \overline{s}_i^{c_i}, \overline{\mathtt{dec}}_i^{c_i}) = 1$, $\mathsf{Com}(\overline{s}_i^{1-\overline{c}_i}; \overline{R}_1^{1-\overline{c}_i}) = \overline{\mathtt{com}}_i^{1-\overline{c}_i}$ and $\beta_0 = \overline{s}_i^0 \oplus \overline{s}_i^1$.
   - The transcript $(\pi_{i,1}, \pi_{i,2}, \pi_{i,3})$ is accepting for $\mathcal{V}_{\mathsf{AI}}$ for the theorem $(x, a, \alpha, \{\mathsf{views}'_i\}_{i\in[\lambda]})$, where x and views'_i are defined as before.
   - $\mathcal{R}^\Sigma$ accepts the commitment $(\pi^\sigma_{i,1}, \pi^\sigma_{i,2}, \pi^\sigma_{i,3})$.

Decommitment procedure $\mathsf{Dec}_{\mathsf{nmc}}$: This proceeds as follows.

1. $\mathcal{S}_{\mathsf{nmc}}$ sends the decommitment information for each of the λ executions of Σ for the message $(m||r_1||, \ldots, ||r_\ell)$: $\{\mathtt{dec}^\sigma_{i,j}\}_{i\in[\lambda], j\in[n]}$ and $\{\mathsf{view}^\sigma_{i,j}\}_{i\in[\lambda], j\in[n]}$.
2. $\mathcal{R}_{\mathsf{nmc}}$ checks majority of the decommitment information for Σ are valid w.r.t. the message $(m||r_1||, \ldots, ||r_\ell)$, and accepts m as the decommitted message if $(m||r_1||, \ldots, ||r_\ell)$ is consistent with $\{a_i, \alpha_i\}_{i\in[\ell]}$.

Fig. 6: $\Pi_{\mathsf{nmc}} = ((\mathcal{S}_{\mathsf{nmc}}, \mathcal{R}_{\mathsf{nmc}}), \mathsf{Dec}_{\mathsf{nmc}})$

- A 3-round public-coin delayed-input *adaptive-theorem* NMZKC (for a three-round extractable commitment) commit-and-prove protocol for the relation $\mathsf{Rel}_{\mathsf{tr}} = \{((m_0, m_1), w) : m_0 = w \vee m_1 = w\}$. We denote the NMZKC simulator with $\mathsf{Sim}_{\mathsf{tr}}$, and recall that the input of $\mathsf{Sim}_{\mathsf{tr}}$ requires only the challenge to compute the first round, where the challenge has the following structure (α, π_2), with $\pi_2 = (\{\pi_{i,2}\}_{i\in[\lambda]}, \{\pi^{\sigma}_{i,2}\}_{i\in[\lambda]})$, where $\{\pi_{i,2}\}_{i\in[\lambda]}$ denotes λ challenges, one for each execution of Π_{AI}, and $\{\pi^{\sigma}_{i,2}\}_{i\in[\lambda]}$ represents the challenge for the λ executions of the Σ-commitment parametrized with $n = 8, k = 2$. Without loss of generality, we assume that $m = |\pi_{i,2}| = |\{\pi^{\sigma}_{i,2}\}_{i\in[\lambda]}| = |\alpha|$.
- The Σ-commitment $\Sigma = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ defined in Figure 3, Section 7.1 parametrized with $n = 8, k = 2$.
- The adaptive-input SHVZK $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ with adaptive-input soundness (and negligible soundness error) for the $\mathcal{NP}$-relation

$$\mathsf{Rel}_{\mathsf{AI}} = \{(x, \{\mathsf{view}_{i_j}\}_{j\in[k]}), (r, \{\mathsf{view}_i\}_{j\in[n]}) : 1 \le i_1 < \cdots < i_k < n \wedge w = \mathsf{Recon}(\{\mathsf{view}_i\}_{j\in[n]}) \ \wedge \ \mathsf{Rel}_{\mathtt{com}}(x, w) = 1\} \quad \textit{where}$$

$$\mathsf{Rel}_{\mathtt{com}} = \left\{ \begin{matrix} x = \left(\{a_i, \alpha_i\}_{i\in[\ell]}\right) \\ w = \left(m, \{r_i\}_{i\in[\ell]}\right) \end{matrix} \middle| \ \forall\ i\ \in [\ell]\ a_i = m + r_i\alpha_i \right\}.$$

- A one-of-two binding commitment scheme $\Pi_{\mathsf{comWI}} = (\mathcal{P}_{\mathsf{comWI}}, \mathcal{V}_{\mathsf{comWI}})$ (Definition 6).

We explicitly require Π_{tr} to be protocol constructed in Section 7.2 because in the security proof we will exploit the structure of the protocol. More detail will follow.

We propose the formal description of our protocol in Figure 6, and prove the following.

Theorem 5. *The protocol $\Pi_{\mathsf{nmc}} = ((\mathcal{S}_{\mathsf{nmc}}, \mathcal{R}_{\mathsf{nmc}}), \mathsf{Dec}_{\mathsf{nmc}})$, described in Figure 6 is a 4-round non-malleable commitment.*

Proof. Binding. The binding property follows from the adaptive-input soundness of Π_{CP} and Π_{AI}, and the binding property of the underlying Σ used to implement Π_{wnmc}.
Non-Malleability. We denote by $\{\mathsf{mim}^{\mathcal{A}}_{H^m_i}(z)\}_{z\in\{0,1\}^\lambda}$ the random variable describing the view of the MiM $\mathcal{A}$ combined with the values that $\mathcal{A}$ commits in the right session in hybrid $H^m_i(z)$.

As required by the definition, we need to show that the distribution of the messages committed by the MiM (together with its view) when receiving an honestly generated commitment of m_0 on the left session and the distribution of the messages committed in the right session by the MiM (together with its view) when computing on the left session an honestly generated commitment of m_1, are indistinguishable.

We proceed via hybrid experiments:

$H^{m_b}_1$**:** In this hybrid in the left session $\mathcal{S}_{\mathsf{nmc}}$ commits to m_b, while in the right session $\mathcal{R}_{\mathsf{nmc}}$ interacts with $\mathcal{A}$. In Claim 3 we prove that in $H^{m_b}_1$ if $\mathcal{A}$ that provides an accepting transcript in the right session, then he does not commit (unless with negligible probability), to $m = \perp$. That is, if the receiver accepts a transcript during the commitment phase, then such commitment admits a valid opening.

Claim 3 *Let $\bar{p}$ be the probability that in the right session of $H^{m_b}_1$ $\mathcal{A}$ successfully commits to a message $\tilde{m} = \perp$, then $\bar{p} < \nu(\lambda)$ for some negligible function ν, for any message $m_b \in \{0,1\}^\lambda$.*

Proof. Let us assume by contradiction that Claim 3 does not hold, then in the right session $\mathcal{A}$ commits to $\perp$. Formally, this means that the adversary is not committing in the majority of the Σ-commitments to the same message $(m, r_1, \ldots, r_\ell) = w$. In particular, the adversary can do that by 1) computing the majority of the executions of the Σ-commitments in an ill-formed manner and 2) using different messages for different execution of the Σ-commitments.

We start by proving that if the adversary provides an accepting transcript for the non-malleable commitment with some non-negligible probability p, then the majority of the Σ-commitments are well formed. We have already argued in the PoK proof of Theorem 4, that if the challenge $\pi_2^\sigma = \{\pi_{i,2}^\sigma\}_{i\in[\lambda]}$ is randomly generated, then it must be that the majority of the Σ-commitments are well formed. We note that in our protocol, π_2^σ is equal to $\beta_0 \oplus \beta_1$, where β_0 is committed using using the ambiguous commitment (β_0 is committed λ times).

This means that $\beta_0 \oplus \beta_1$ is not fully under the control of the verifier (hence, under the control of the PoK extractor) unless we can argue there exists j such that $(\overline{\mathtt{com}}_j^b, \overline{\mathtt{dec}}_j^b) \leftarrow \mathsf{Com}(\overline{s}_b; \cdot)$. That is at least one pair of ambiguous commitments used to commit to β_0 such that both are computed using the honest procedure. Assume by contradiction that such j does not exist, then the only way the adversary has to provide an accepting transcript is by either guessing the challenge $\overline{\gamma}_1$, and/or by equivocating $\overline{\gamma}_0$ accordingly to the value γ_1 received by the honest receiver.

Given that the probability that the first event happens is negligible (in particular, it is not possible for the adversary to guess more than $\lambda/3$ bits[13] of the challenge $\overline{\gamma}_1$), it must be that the adversary provides an accepting transcript by programming $\overline{\gamma}_0$ accordingly to $\overline{\gamma}_1$. However, if this happens, then, following the proof of Theorem 4, we can design an extractor that extracts the value committed in one of the Σ-commitment of Π_{tr} with non-negligible probability.

We now argue that the value extracted corresponds, with non-negligible probability, to either the message $\hat{s}_0$ or the message $\hat{s}_1$ the verifier sends in the clear in the right session as part of the third round of the one-of-two binding commitment scheme.

To argue this we can simply rely on the soundness of the protocol Π_{AI} that is run inside Π_{tr}. Indeed, note that the challenge for a single execution of Π_{tr} corresponds to $\overline{\gamma}_0$.

We are now ready to conclude the first part of the claim's proof, by showing how to use such an adversary to break the hiding of the one-of-two binding commitment scheme, thus reaching a contradiction. Let $\mathsf{Extractor}$ be the extractor we mentioned that returns either $\hat{s}_0$ or $\hat{s}_1$. The reduction acts as the honest verifier would do, but it will act as a proxy for all the messages related to Π_{comWI} between the adversary and the external challenger. The extractor could return either $\hat{s}_0$ or $\hat{s}_1$. However, given that the first round of Π_{comWI} can depend only on either of the two messages, it must be that the extractor returns the value committed in the first round of Π_{comWI}. Hence, such an adversary would break the equivocability property of Π_{comWI}.

We mentioned at the beginning of the proof, that the other way the adversary can commit to an ill-formed (but accepting) commitment, is by committing to different values in different execution of the Σ-commitment. Note that the soundness of $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ and the way the values $a_1, \ldots, a_\ell$ are computed in the last round of the protocol should guarantee that all the well-formed execution of the sigma-commitment commits to the same w. However, we cannot trivially rely on the soundness of Π_{AI}, since the adversary might compute the first round of Π_{AI} adaptively on the challenges $(\pi_{2,i})$ it receives. The adversary could do that by relying

[13] We adversary cannot guess more than a small fraction of λ bits, but for this proof it is sufficient to use this upper bound.

(again) on the equivocation property of the ambiguous commitment scheme Π_{com}. We can argue that with non-negligible probability, at least on the executions of the ambiguous are computed in a non-binding mode. This would allow us to use rely on the binding of Π_{AI}. Suppose by contradiction that all the ambiguous commitments are computed in equivocal mode. The only way the adversary has to compute an accepting transcript is by opening γ_0 adaptively on γ_1. In particular, the adversary needs to do that for at least $\lambda/2$ bits of γ_0 (otherwise the adversary would be caught cheating with overwhelming probability). In this case, we can again rely on the same argument as above since this adversary just corresponds to an adversary that provides accepting transcripts for the protocol Π_{tr} with respect to sufficiently many random challenges (specified by γ_0). Hence Extractor would again return the trapdoors.
Note in particular, that Extractor can detect whether $\mathcal{A}$ is performing a commitment of $\perp$. We note that if $\mathcal{A}$ is committing to a message $\tilde{m} \neq \perp$, Extractor will not return the trapdoor. In this case, we can run an additional extraction process that instead returns the message $\tilde{m}$ with non-negligible probability. Such an extractor simply rewinds from the fourth to the second round sending a fresh third round in the right session thus obtaining $(\tilde{a}_1, \ldots, \tilde{a}_\ell)$ from the main thread, and $(a'_1, \ldots, a'_\ell)$ from the rewinding thread. The extractor can then interpolate the points to get the committed message $\tilde{m}$. We are now ready to claim the following. We denote with $\mathsf{Extractor}'$, the extractor that first runs Extractor to check whether $\mathcal{A}$ committed to $\perp$ (i.e., it checks whether Extractor returns a trapdoor), and if this is not the case, the extractor runs the procedure we have just mentioned to extract the message committed by $\mathcal{A}$. Due to the above, we can claim the following.

Claim 4 *If in $H_1^{m_b}$ the adversary provides an accepting transcript with non-negligible probability, then there exists a* PPT *extractor* $\mathsf{Extractor}'$ *that returns the committed message, if there exists one, or it returns $\perp$ if the commitment does not admit a valid opening.*

$H_2^{m_b}$**:** This hybrid is identical to the previous one except that the trapdoor witness is extracted and used in both the executions of Π_{tr}. In more detail, $\mathcal{A}$ is rewound from the 3rd to the 2nd round in the left session in order to obtain a second third round. Let $\hat{s}_0^1, \hat{s}_1^1$ be the strings that $\mathcal{A}$ sent in the third round before being rewound. Then, the hybrid computes the following steps:

a) The hybrid repeats the following until receives a new third round from $\mathcal{A}$ or $\mathsf{nr}(\lambda)$-trials are executed.
- Execute again the 2nd round of Π_{nmc} sampling a new challenge for Π_{comWI}, and committing to $\hat{s} = \hat{s}_0^1$ using the honest prover procedure in both the executions of Π_{tr}. Upon receiving a 3rd round from $\mathcal{A}$, let $\hat{s}_0^2, \hat{s}_1^2$ be the strings that she send. If $\hat{s}_0^2 = \hat{s}_0^1$ the hybrid completes the left session as acting as described in $H_1^{m_b}$. If $\hat{s}_1^2 \neq \hat{s}_1^1$ stop returning $\perp$. Otherwise, step b is computed.

b) The hybrid repeats the following until receives a new third round from $\mathcal{A}$ or $\mathsf{nr}(\lambda)$-attempts are executed.
- Execute again the 2nd round of Π_{nmc} sampling a new challenge for Π_{comWI}, and committing to $\hat{s} = \hat{s}_1^1$ using the honest prover procedure in both the execution of Π_{tr}. Upon receiving a 3rd round from $\mathcal{A}$ the hybrid completes the left session as acting as described in $H_1^{m_b}$.

c) If $\mathsf{nr}(\lambda)$-trials are already executed stop and return $\perp$.

Let p be the probability that $\mathcal{A}$ provides an accepting third round in the left session, following the arguments of [KOS18] (that in turn are based on [GK96]) we can argue that our simulator

succeeds with probability $p - \mathsf{negl}(\lambda)$ in extracting the message $\hat{s}$ in expected polynomial time. Moreover, due to the binding of Π_{comWI}, we can claim that we have extracted the value committed in the first round by the adversary (which appears in the last round of Π_{comWI} of all the rewinding threads).

We now argue that if the adversary distinguishes between the two hybrids, then we can construct an adversary that breaks the property of NMZKC of Π_{tr}. We note that the reduction has to be strictly polynomial time, while the extraction of $\hat{s}$ takes a number of steps that are polynomial in expectation. Therefore, we consider a truncated experiment in which we set the number of attempts to extract $\hat{s}$ to be $(\lambda \cdot p_{\mathsf{dis}}(\lambda) \cdot \mathsf{nr}(\lambda))$. By an averaging argument, we can show that the in the truncated experiments we manage to extract $\hat{s}$ with non-negligible probability (we recall that here by contradiction we are assuming that an adversary distinguishes between the two hybrids with some non-negligible probability). Our reduction to the NMZKC property of Π_{tr} works as follows.

Left session. In the main thread, and during the rewinding threads, the reduction computes the messages of the protocol as in $H_1^{m_b}$. If the extraction is not successful, then the reduction returns a random bit, else it sends to the external challenger the second-rounds for the two instantiations of Π_{tr}, respectively, $\pi_2^{\mathsf{tr}} = (\overline{\gamma}_0, \overline{\gamma}_0, \overline{\gamma}_0^2)$ and $\overline{\pi}_2^{\mathsf{tr}} = (\gamma_0, \gamma_0, \gamma_0^2)$, and the witness $\hat{s}$ (we recall that the NMZKC simulator of Π_{tr} is adaptive only in the theorem, hence, the witness needs to be specified at before the first round is computed). The reduction, upon receiving π_1^{tr} and $\overline{\pi}_1^{\mathsf{tr}}$, acts exactly as in $H_1^{m_b}$ to compute the remaining messages that constitute the second round of the protocol. Upon receiving $(\hat{s}_0, \hat{s}_1)$ the reduction forwards these two values to the challenger (this pair represents the theorem for Π_{tr}). Note that it must be that $\hat{s}_0 = \hat{s}$ or $\hat{s}_1 = \hat{s}$ with the same probability both in $H_1^{m_b}$ and $H_2^{m_b}$ otherwise we can already distinguish between whether the challenger is computing the messages of Π_{tr} using the simulated or the honest procedure. The reduction, upon receiving $(\pi_3^{\mathsf{tr}}, \overline{\pi}_3^{\mathsf{tr}})$ from the challenger, computes the fourth round as in $H_1^{m_b}$, except that it uses π_3^{tr} and $\overline{\pi}_3^{\mathsf{tr}}$ to compute the fourth round of the right session.

Right session. In the right session, the reduction acts as the honest receiver would do.

The output of the reduction corresponds to the output of the distinguisher for the two hybrid experiments on input the view of $\mathcal{A}$ and to the message returned by $\mathsf{Extractor}'$.

We recall that in Claim 3 we have proven that in the previous hybrid $\mathcal{A}$ does not commit to $\perp$, moreover, in Claim 4 we have proven that we can detect whether $\mathcal{A}$ is performing a commitment of a valid message, and if that is the case, extract such a message by using $\mathsf{Extractor}'$. We recall that the definition of non-malleable HVZK with respect to commitment ensures that the distribution of the messages committed via an extractable commitment that is run in parallel with Π_{tr} is independent of whether Π_{tr} is computed using the honest prover procedure, or the simulated procedure. The commitment that is run in parallel with Π_{tr} simply corresponds to the commitment computed by $\mathcal{A}$ on the right session. This commitment is extractable, indeed we have argued that $\mathsf{Extractor}'$ returns the message committed by $\mathcal{A}$ in the right session. Note that if the challenger has computed the messages of Π_{tr} using the honest prover procedure, then the output of the reduction corresponds to the output of the adversary in $H_2^{m_b}$, else it corresponds to the output of the adversary in $H_1^{m_b}$. The property of NMZKC guarantees that the distribution of the committed message on the right session does not change. From the above it follows that $\{\mathsf{mim}^{\mathcal{A}}_{H_2^{m_b}}(z)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{mim}^{\mathcal{A}}_{H_1^{m_b}}(z)\}_{z\in\{0,1\}^\lambda}$. Moreover, form the above and from Claim 3 we have the following

Claim 5 *Let $\bar{p}$ be the probability that in the right session of $H_2^{m_b}$ $\mathcal{A}$ successfully commits to a message $\tilde{m} = \perp$, then $\bar{p} < \nu(\lambda)$ for some negligible function ν, for any message $m_b \in \{0,1\}^\lambda$.*

$H_3^{m_b}$**:** This hybrid is equal to the previous, with the exception that the ambiguous commitments for which the receiver does not ask the randomness (that would allow to check whether a commitment is computed using the trapdoor or the honest procedure) are computed using the equivocal procedure. In particular, the hybrid selects random $c, \bar{c}$, and computes $\mathtt{com}_i^{c_i\prime} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^\ell; r_i^{c_i})$, $\overline{\mathtt{com}}_i^{\bar{c}_i\prime} \leftarrow \mathsf{Com}^{\mathsf{eq}}(1^\ell; \overline{r}_i^{\bar{c}_i})$ for all $i \in [\lambda]$. Upon receiving $\gamma_1, \overline{\gamma}_1$, the hybrid computes $\gamma_1 \oplus c = \gamma_0$, $\overline{\gamma}_1 \oplus \bar{c} = \overline{\gamma}_0$, and computes an accepting transcripts $(\pi_1^{\mathsf{tr}}, (\gamma_0, \gamma_0, \gamma_0^2), \pi_3^{\mathsf{tr}}))$ $(\overline{\pi}_1^{\mathsf{tr}}, (\overline{\gamma}_0, \overline{\gamma}_0, \overline{\gamma}_0^2), \overline{\pi}_3^{\mathsf{tr}}))$ for Π_{tr} with respect to the challenge γ_0 (reps. $\overline{\gamma}_0$). Note that we can do that since Π_{tr} is computed using the honest prover procedure, hence, we can compute an accepting transcript for any possible challenge received by the MiM, while keeping fixed c and $\bar{c}$. We now prove that if the distribution of the committed message changes between the two hybrids we can make a reduction to the equivocability property of ambiguous commitments. The reduction acts on the left session exactly as $H_3^{m_b}$, with the exception that the for each $i \in [\lambda]$, for a randomly chosen bits c_i and $\bar{c}_i$, the commitments $\mathtt{com}_i^{c_i\prime}$ and $\overline{\mathtt{com}}_i^{\bar{c}_i\prime}$ is generated by an external challenger. Moreover, upon receiving $\gamma_1, \overline{\gamma}_1$, the reduction computes $c \oplus \gamma_1 = \gamma_0$, $\bar{c} \oplus \overline{\gamma}_1 = \overline{\gamma}_0$ and computes accepting transcripts of Π_{tr}, with respect to the challenge γ_0 and $\overline{\gamma}_0$. Note that reduction can do that since we extract a valid witness to execute Π_{tr} using the honest prover procedure.
On the right session instead, the reduction simply runs $\mathsf{Extractor}'$. We note that the rewinds performed by $\mathsf{Extractor}'$ do not perturb the reduction. We also observe that the extractor must successfully extract the committed message since a failure in the extraction can already be used to distinguish between the choice bit of the challenger of the ambiguous commitment. Note that nothing prevents $\mathcal{A}$ to compute some of the ambiguous commitments using the equivocal procedure in $H_3^{m_b}$, but during the rewinds, the adversary must open at least one of these commitments always to the same value during the rewinds (exactly as we do on the left session of the reduction and of $H_3^{m_b}$). If the adversary does not do that, this makes the two hybrids immediately distinguishable.
Now that we have argued that the probability of success of $\mathsf{Extractor}'$ does not depend on the choice bit of the challenger, we can run the distinguisher using as input the view of the MiM involved in the reduction, and the message extracted via $\mathsf{Extractor}'$. If the distinguisher distinguishes with a non-negligible advantage, then we have constructed a valid adversary for the ambiguous commitment. Indeed, it is easy to see that if the ambiguous commitments are computed using the non-equivocal procedure, then the view of MiM corresponds to $H_2^{m_b}$, else it corresponds to $H_3^{m_b}$.
Therefore $\{\mathsf{mim}^{\mathcal{A}}_{H_3^{m_b}}(z)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{mim}^{\mathcal{A}}_{H_2^{m_b}}(z)\}_{z\in\{0,1\}^\lambda}$.
From the above and from Claim 5, we have the following claim.

Claim 6 *Let $\bar{p}$ be the probability that in the right session of $H_3^{m_b}$ $\mathcal{A}$ successfully commits to a message $\tilde{m} = \perp$, then $\bar{p} < \nu(\lambda)$ for some negligible function ν, for every message $m_b \in \{0,1\}^\lambda$.*

$H_4^{m_b}$**:** This hybrid is equal to the previous with the exception that, for each $i \in [\lambda]$, the messages $\pi_{i,1}, \pi_{i,3}$ are generated using the SHVZK simulator $\mathsf{Sim}_{\mathsf{AI}}$. If by contradiction the distribution of the committed message changes in this hybrid, then we can make a reduction to the adaptive SHVZK of Π_{AI}.

To construct the reduction we distinguish between two types of schedules: synchronous schedule and asynchronous. We focus on the synchronous schedule first. Let $\mathsf{CH}^{\mathsf{shvzk}}$ be the challenger for the adaptive-input SHVZK of Π_{AI}.

Then, we can construct an adversary $\mathcal{A}^{\mathsf{shvzk}}$ that interacts with $\mathcal{A}$ in the left and the right session according to both $H_3^{m_b}$ and $H_2^{m_b}$ for all messages except for the messages of Π_{tr}. For these messages the reduction acts as a proxy between $\mathcal{A}$ and $\mathsf{CH}^{\mathsf{shvzk}}$ in the left session. More formally, the reduction $\mathcal{A}^{\mathsf{shvzk}}$ proceeds as follow:

Left session.

1. Upon receiving the 1st round from $\mathcal{A}$, apply the trapdoor extraction procedure, thus obtaining the string $\hat{s}$ (in this step the reduction truncates the running time as explained in the previous reductions).
2. Obtained $\hat{s}$, in the left session act exactly as in $H_4^{m_b}$, but compute the messages related to Π_{AI} as follows.
3. For each $i \in [\lambda]$ let $\pi^{\sigma}_{i,3}$ be the third round of Σ obtained by executing $\mathcal{S}^{\Sigma}$ on input $\pi^{\sigma}_{i,2}$ (note that $\pi^{\sigma}_{i,3} = \{\mathsf{dec}^{\sigma}_{i,j}, \mathsf{view}^{\sigma}_{i,j}\}_{j \in \pi^{\sigma}_{i,2}}$), define $\mathsf{views}'_i \leftarrow \{\mathsf{view}^{\sigma}_{i,j}\}_{j \in \pi^{\sigma}_{i,2}}$ and $\mathsf{views}_i \leftarrow \{\mathsf{view}^{\sigma}_{i,j}\}_{j \in [n]}$.
4. Define the pair statement-witness $((x, \{\mathsf{views}'_u\}_{u \in \lambda}), \{\mathsf{views}_u\}_{u \in \lambda})$ and send it to $\mathsf{CH}^{\mathsf{shvzk}}$ together with $\{\pi_{i,2}\}_{i \in [\lambda]}$
5. Upon receiving $\{\pi_{i,1}, \pi_{i,3}\}_{i \in [\lambda]}$ from $\mathsf{CH}^{\mathsf{shvzk}}$, for each $i \in [\lambda]$ equivocate the i-th ambiguous commitment in the pair, to a share that reconstructs $\pi_{i,1}$.
6. Complete the right execution exactly as in $H_4^{m_b}$ but using the messages $\{\pi_{i,1}, \pi_{i,3}\}_{i \in [\lambda]}$ when needed.
7. After the main thread is completed, for any rewind performed on the right session by $\mathsf{Extractor}'$ that requires completing a new transcript for Π_{AI}, the reduction acts exactly as in $H_3^{m_b}$. In particular, all the messages for Π_{AI} are computed using the honest prover procedure. That is, the ambiguous commitments are opened always to the same shares that lead to the set of values $\{\overline{\pi}_{i,1}\}_{i \in [\lambda]}$ computed using the honest prover procedure of Π_{AI} (i.e., in the rewinding thread it is used always the same set of first rounds).

Right session.

Upon completion of the main thread run $\mathsf{Extractor}'$. Upon receiving $(\tilde{m}, \tilde{r}_1, \ldots, \tilde{r}_\ell)$ from the extractor, checks whether the values $(\tilde{a}_1, \ldots, \tilde{a}_\ell)$ $(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_\ell)$ generated in the right session of the main thread are consistent with $(\tilde{m}, \tilde{r}_1, \ldots, \tilde{r}_\ell)$ and $(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_\ell)$, if this is the case then return $\tilde{m}$, else return $\perp$.

The reduction runs the distinguisher for the two hybrids on input in the view of the MiM and the value extracted from $\mathsf{Extractor}'$ and returns what the distinguisher returns.

We observe that during the extraction phase, the reduction in the left session acts exactly as $H_3^{m_b}$, and we have proven that $\mathcal{A}$ in $H_3^{m_b}$ commits correctly to a message $\tilde{m} \neq \perp$. Hence, we can check whether this message is consistent with the message committed in the main thread, where the messages of Π_{AI} are computed by the external challenger. We note that if the probability that this new extraction procedure fails differs between the two hybrids, then this already creates a distinguishing advantage that can be used to break the security of Π_{AI}. Hence, we can claim that the message extracted in our reduction is the correct one. This part of the proof ends with the observation that if the challenger computes messages for Π_{AI} using the SHVZK, then the view of the adversary corresponds to the one in $H_4^{m_b}$, otherwise, it corresponds to the view in $H_3^{m_b}$

The proof for the asynchronous case follows similar (simpler) arguments given that the rewinds performed by $\mathsf{Extractor}'$ trivially do not affect the reduction. Therefore $\{\mathsf{mim}^{\mathcal{A}}_{H_4^{m_b}}(z)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{mim}^{\mathcal{A}}_{H_3^{m_b}}(z)\}_{z\in\{0,1\}^\lambda}$.
From the above and from Claim 6 we can claim the following

Claim 7 *Let $\bar{p}$ be the probability that in the right session of $H_4^{m_b}$ $\mathcal{A}$ successfully commits to a message $\tilde{m} = \perp$, then $\bar{p} < \nu(\lambda)$ for some negligible function ν, for every message $m_b \in \{0,1\}^\lambda$.*

H_5**:** This hybrid is acting as the previous with the difference that Π_{wnmc} to commit to the message m_{1-b}, in particular, $\mathcal{S}^{\Sigma}$ is executed w.r.t. the message $(m_{1-b}, r_1, \ldots, r_\ell)||r)$ and $a_i \leftarrow r_i\alpha_i + m_{1-b}$. Note that this hybrid corresponds to $H_4^{m_{1-b}}$.
Suppose by contradiction that $H_4^{m_{1-b}}$ is distinguishable from H_5, then there exists a distinguisher that breaks the security of Π_{wnmc}. In the case the schedule is synchronous, then we can simply rely on the weak-non-malleability of Π_{wnmc}. We recall that weak-non-malleability guarantees that Π_{wnmc} remains non-malleable as long as the MiM adversary provides a well-formed transcript for Π_{wnmc}. Given that we have proven that in $H_4^{m_b}$ and in $H_4^{m_{1-b}}$ $\mathcal{A}$ does not commit to a message $\tilde{m} = \perp$, we can claim that $\{\mathsf{mim}^{\mathcal{A}}_{H_5}(z)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{mim}^{\mathcal{A}}_{H_4^{m_b}}(z)\}_{z\in\{0,1\}^\lambda}$ due to the weak non-malleability of Π_{wnmc}.
In the case of an asynchronous schedule, we can rely on the hiding of Π_{wnmc}. The reduction would act in the left session as a proxy between the challenger of the hiding game for Π_{wnmc} and $\mathcal{A}$. After that the MiM completes its commitment, the reduction runs $\mathsf{Extractor}'$ to extract the message committed on the right session. The reduction now can input $\mathcal{A}$'s view and the extracted message to the distinguisher of the two hybrids, and returns whatever the distinguisher returns.

The proof ends with the observation that

$$\{\mathsf{mim}^{\mathcal{A}}_{H_1^{m_0}}(z)\}_{z\in\{0,1\}^\lambda}, \approx \cdots \approx \{\mathsf{mim}^{\mathcal{A}}_{H_4^{m_0}}(z)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{mim}^{\mathcal{A}}_{H_5}(z)\}_{z\in\{0,1\}^\lambda} = \{\mathsf{mim}^{\mathcal{A}}_{H_4^{m_1}}(z)\}_{z\in\{0,1\}^\lambda} \approx \ldots$$
$$\cdots \approx \{\mathsf{mim}^{\mathcal{A}}_{H_1^{m_1}}(z)\}_{z\in\{0,1\}^\lambda}.$$

□

9 Comparison with Previous Non-Black-Box Approaches to Four-Round Non-malleable Commitments.

As we argued, our main strategy to construct a non-malleable commitment scheme is to lift the security of the weak non-malleable commitment scheme of [GRRV14b, Fig. 2] (that we also recall in Figure 5), relying on a special notion of zero-knowledge that we call non-malleable HVZK with respect to commitment. This notion guarantees that a sender of a commitment scheme does not change the distribution of the committed messages depending on whether he receives an honestly generated zero-knowledge proof or a simulated one. We construct a NMZKC for a specific class of commitments, which includes the weak-non-malleable commitment scheme of [GRRV14b, Fig. 2].

Our approach is inspired by [GRRV14b], where the authors also lift the security of a weak-non-malleable commitment scheme relying on zero-knowledge. However, our techniques significantly depart from those of [GRRV14b]. In the next paragraphs, we highlight the main difference between the two approaches and explain why we could use as one of the main building block the simple

weak-non-malleable commitment of [GRRV14b, Fig. 2], instead of a modified version, as the authors of [GRRV14b] do.

The main technical challenge in designing non-malleable commitments with low round complexity is due to arguing in the proof that the security of the primitives involved in the protocol is maintained despite performing rewinds to extract the message committed by the MiM (on the right session). One of the primitives involved in the scheme of Goyal et al. is a non-rewind secure witness-indistinguishable proof denoted by Π. To cope with the rewinds performed by the extractor in the proof (while still relying on the WI property of Π), Goyal et al. adopt the following approach. The prover prepares n first rounds for the non-rewind secure WI protocol (denoted by Π). Upon receiving one valid second round from the verifier, the prover picks one instance of Π at random (let us say the i-th) and completes the proof providing an accepting third round only with respect to the i-th instance. Let us denote the above protocol by Π_{rew}.

Despite this protocol being rewind secure, Goyal et al. cannot use just one execution of Π_{rew}, which proves that either the committer has behaved honestly in the algebraic part of the commitment or that the committer knows a trapdoor. Indeed, there is a simple adversarial strategy for which the proof of [GRRV14b] would not work in this case. Intuitively, consider a MiM that completes an execution on the right session only if it receives a proof for the j-th instance of Π, and aborts in any other case (note that this MiM is non-aborting with non-negligible probability). This MiM would make the reduction to the WI of Π fail. In particular, any rewind performed by the extractor on the right session would make the MiM ask different second rounds for the same execution of Π (or abort if on the left session a different instance of Π is completed). To solve this problem Goyal et al. compute a secret sharing of the message and perform one execution of Π_{rew} for each of the shares. Now, even if the MiM applies the same strategy to one run of Π_{rew}, it is safe to allow the MiM to perform this rewind since the only thing that will be leaked is a share of the message m (note that two accepting transcripts for the same execution of Π for two different second rounds might completely leak the witness). In the formal proof, Goyal et al. need to rely on the fact that the number of executions of Π that are not rewound (and consequently the number of shares not leaked) is sufficient to protect the secrecy of the message m. This modification also requires changing how the extractor works (e.g., by relying on the quadratic polynomials). Hence, to obtain their non-malleable commitment scheme, the authors of [GRRV14b] rely on a more sophisticated version of the weak-non-malleable commitment proposed in their work.

In our paper, we do not rely on any rewind secure primitive (which we replace with a proof system non-malleable with respect to commitments), so we do not need to split the message into shares and follow the strategy described above. We note that similarly to us, also [COSV17a] relies on the simpler sub-scheme of [GRRV14b, Fig. 2] to obtain a 4-round concurrent non-malleable commitment scheme. To summarize, the main difference between ours and the [GRRV14b] approach (that relies on rewind secure primitive) is that our work is based on the observation that the rewinds are performed in the reductions or during the simulation, and as such, the adversary does not have clue that the rewinds are happening. Hence, relying on primitives that are rewind-secure (i.e., the adversary can consciously make rewinds and collect the transcripts generated during the rewinds) can be avoided for the application we consider in the paper.

10 Our Concrete Instantiation with BMR

In this section we give our instantiation of a robust MPC protocol according to Definition 16. Our protocol Π_{RobBMR} is a BMR-style [BMR90] protocol: it consists of two main steps, a multiparty garbling $\Pi^{\mathtt{off}}_{\mathsf{RobBMR}}$ (Figure 7) and an online computation which we shortly describe.

We consider binary circuits C_f consisting of $|C_f|$ gates, each of which has two input wires, u and v, and one output wire w. We use g to indicate both the gate index and the gate function. Let W be the set of all wires in the circuit, W_{in} and $W_{\mathtt{out}}$ be the set of input and output wires, respectively; we denote by W_{in_i} the set of input wires associated to party $\mathtt{P}_i$.

We recall that any garbled-circuit based protocol is a two-phase protocol consisting in an input-independent phase, also called *garbling*, and an online evaluation.

Garbling Pre-processing. In this phase all parties $\mathtt{P}_1, \ldots, \mathtt{P}_n$ involved in the protocol generate a sharing of the garbled circuit according to some fixed secret sharing scheme $\langle\cdot\rangle$ with t_p-privacy. As in any other garbled-circuit based protocol, to garble a Boolean circuit each wire is assigned two random keys $\boldsymbol{k}_{w,0}, \boldsymbol{k}_{w,1}$ encoding the 0-value and 1-value, respectively. The goal of the process is to generate four ciphertexts for each gate according to the gate function, such that each output-wire key is encrypted according to all combinations of input-wire keys which evaluate to that output wire key.

More in particular, in multiparty garbling each party $\mathtt{P}_i$ samples two random keys $\boldsymbol{k}^i_{w,0}$ and $\boldsymbol{k}^i_{w,1}$ and a random wire mask $\lambda^i_w \in \{0,1\}$, for each wire $w \in W$. Given wire masks $\lambda_u, \lambda_v, \lambda_w$ and wire keys $\{\boldsymbol{k}^i_{u,\alpha}, \boldsymbol{k}^i_{v,\beta}, \boldsymbol{k}^i_{w,0}, \boldsymbol{k}^i_{w,1}\}_{(\alpha,\beta)\in\{0,1\}^2, i\in[n]}$, parties generate a garbled gate corresponding to the gate truth table, as follows. It consists of four rows, indexed by the values $(\alpha,\beta) \in \{0,1\}^2$ on the input wires, where every row contains n ciphertexts, each of which is encrypted under $2n$ keys as follows:

$$\langle \tilde{g}^j_{\alpha,\beta} \rangle = \langle \left(\bigoplus_{i=1}^{n} (F_{\boldsymbol{k}^i_{u,\alpha}}(g\|\beta\|j) \oplus F_{\boldsymbol{k}^i_{u,\beta}}(g\|\alpha\|j)) \right) \oplus \boldsymbol{k}^j_{w,0} \oplus \chi_{g,\alpha,\beta} \cdot (\boldsymbol{k}^j_{w,0} \oplus \boldsymbol{k}^j_{w,1}) \rangle, \tag{1}$$

where $j \in [n]$, and it represents the j-th ciphertext on the (α,β)-row, $\chi_{g,\alpha,\beta} = g(\lambda_u \oplus \alpha, \lambda_v \oplus \beta) \oplus \lambda_w$ and F is a pseudo-random function (PRF).

Online Evaluation. During the online evaluation, these encrypted truth tables, along with the circuit-output wire masks, are revealed to all parties so to allow local evaluation of the circuit. More precisely, the two-round BMR online step proceeds as follows.

1. For every wire w, which is the circuit-input wire of party $\mathtt{P}_i$, each party $\mathtt{P}_i$ broadcasts values $\Lambda_w = \rho_w \oplus \lambda_w$, for each $w \in W_{\mathsf{in}_i}$, where ρ_w is the actual input and λ_w the corresponding wire mask.
2. In response, for every Λ_w received, each party $\mathtt{P}_j$ broadcasts their key $\boldsymbol{k}^i_{w,\Lambda_w}$ corresponding to the publicly known value Λ_w.

Upon collecting all the keys and masked inputs, parties can start evaluating the circuit. At this point, this does not require any interaction. Given a complete tuple of input keys $(\boldsymbol{k}^1_{u,\Lambda_u}, \ldots, \boldsymbol{k}^n_{u,\Lambda_u})$ and $(\boldsymbol{k}^1_{v,\Lambda_v}, \ldots, \boldsymbol{k}^n_{v,\Lambda_v})$, it is possible to decrypt each gate, by computing all the PRFs, obtaining all the corresponding output-wire keys $(\boldsymbol{k}^1_{w,\Lambda_w}, \ldots, \boldsymbol{k}^n_{w,\Lambda_w})$, for each $w \in W \setminus W_{\mathsf{in}}$.

Note that during this evaluation each party decrypts the entire row, requiring n^2 PRF evaluations. Once these output keys are obtained, every party $\mathtt{P}_i$ can check that the i-th key corresponds to

one of its keys $\boldsymbol{k}^i_{w,0}, \boldsymbol{k}^i_{w,1}$ generated in the garbling phase. This check allows: 1) To determine the masked output value, i.e. if $\boldsymbol{k}^i_{w,\epsilon_w} = \boldsymbol{k}^i_{w,0}$, $\mathtt{P}_i$ sets $\Lambda_w = 0$, and $\Lambda_w = 1$ otherwise; 2) To ensure security against any static malicious adversary for the online evaluation, as it allows the honest parties to detect malicious behaviour in case the output keys of a gate does not contain one of the two keys generated in the preprocessing step for the output wire of that gate.

10.1 Our BMR-style instantiation

Protocol $\Pi^{\mathsf{off}}_{\mathsf{RobBMR}}$

NOTATION: Given a gate g, we denote by u (resp. v) its left (resp. right) input wire, and by w its output wire. Let $W_{\mathsf{out}}, W_{\mathsf{in}}$ be the set of output and input wires, respectively, and W_{in_i} the set of input wires for party P_i. Let G be the set of gates in C. Let $F : \{0,1\}^\kappa \times [|G|] \times \{0,1\} \times [n] \to \{0,1\}^\kappa$ be a PRF.
Let $\langle\cdot\rangle$ denote an arbitrary secret sharing, we will specialize it according to our needs, and $\langle\cdot\rangle^i$ be the share corresponding to party P_i.

Generate wire masks and keys: Passing through the wires of the circuit topologically, proceed as follows:

1. Each P_i samples a random $\lambda^i_w \leftarrow \{0,1\}$, and call $\mathcal{F}_{\mathrm{Commit}}$ on input λ^i_w for each output-wire $w \in W_{\mathsf{out}}$.
2. Every P_i samples two keys $\boldsymbol{k}^i_{w,0} \leftarrow \{0,1\}^\kappa$ and $\boldsymbol{k}^i_{w,1} \leftarrow \{0,1\}^\kappa$.
3. Each party P_i, $i \in [n]$, calls $\mathcal{F}_{\mathrm{Commit}}$ on the circuit-input keys $\boldsymbol{k}^i_{w,0}$ and $\boldsymbol{k}^i_{w,1}$, for each $w \in W_{\mathsf{in}_i}$, obtaining commitments $\{\sigma_{i,w,b}\}_{b\in\{0,1\}}$.

Garbling: For each gate $g \in G$, each $j \in [n]$, and the four combinations of $a, b \in \{0,1\}^2$, the parties call $\mathcal{F}_{\mathrm{Garbling}}$ to obtain shares of the j-th entry of the garbled gate $\tilde{g}_{a,b}$ such that

$$\tilde{g}_{a,b} = \left(F_{\boldsymbol{k}^j_{u,a}}(g\|b\|j) \oplus F_{\boldsymbol{k}^j_{v,b}}(g\|a\|j)\right) \oplus \left(\boldsymbol{k}^j_{w,0} \oplus \rho_{j,a,b}\right)$$

where $\rho_{j,a,b} = (\boldsymbol{k}^j_{w,0} \oplus \boldsymbol{k}^j_{w,1}) \cdot \chi_{g,a,b}$ and $\chi_{g,\alpha,\beta} = g(\lambda_u \oplus \alpha, \lambda_v \oplus \beta) \oplus \lambda_w$.

Open garbling : On input (Open), parties reveal their garbled shares and, for every circuit-output wire $w \in W_{\mathsf{out}}$, reveal λ_w to all the parties. Each party P_i broadcast commitments $\{\sigma_{i,w,b}\}_{b\in\{0,1\}}$, for each $w \in W_{\mathsf{in}_i}$

Open committed keys: For each $i \in [n]$ and $w \in W_{\mathsf{in}_i}$, parties call $\mathcal{F}_{\mathrm{Commit}}$ on input commitments $\sigma_{i,w,b}$, where $b \in \{0,1\}$, obtaining the corresponding committed keys.

Fig. 7: BMR preprocessing

Here we describe our instantiation Π_{RobBMR} more concretely.

A complete description of our BMR pre-processing is given in Figure 7. It assumes access to an ideal functionality $\mathcal{F}_{\mathrm{Garbling}}$ that provides the garbled gates shares to the relevant parties, and a standard commitment functionality $\mathcal{F}_{\mathrm{Commit}}$. The pre-processing can be concretely instantiated either with an honest and dishonest majority, similarly to recent efficient black-box BMR preprocessing protocols with active security. The main modification that we require, compared to other BMR-style protocols, are committed circuit-input keys. This will be useful to prove robustness.

Here we prove that our BMR instantiation satisfies the properties required in Section 3.10. Correctness and t_p-privacy properties directly follows by proving the following proposition.

Proposition 1. *Assuming the existence of one-way functions. Let f be an n-party functionality. The protocol Π_{RobBMR}, described in Figure 8, UC-securely computes f in the presence of* static semi-honest adversary *corrupting a set A of t_p parties in the $\mathcal{F}^{\mathsf{off}}_{\mathsf{RobBMR}}$-hybrid model.*

Proof. We start by proving that $\Pi^{\mathsf{on}}_{\mathsf{RobBMR}}$ is perfectly correct. We know that the preprocessing is perfectly correct. In the online phase, the only communication is the broadcasting by each party of

The MPC Protocol - Π_{RobBMR}

INPUTS: A circuit C_f computing the function f, which consists of XOR and AND gates. Let W be the set of all wires in C_f, W_{in_i} be the set of input wires for party P_i, and W_{out} be the set of output wires. Each party P_i has private input $\{w_i\}$. Parties have a public input x.
Let $F : \{0,1\}^\kappa \times [|G|] \times \{0,1\} \times [n] \to \{0,1\}^\kappa$ be a PRF.
The parties execute the following commands in sequence.

Preprocessing: This sub-task is performed as follows.

- Parties call the preprocessing functionality to obtain private keys, a garbled version $\tilde{g}$ of every gate g in C, private wire masks, and commitments $\sigma_{i,w,b}$ on the circuit-input keys, for each $i \in [n]$.

Online Computation: This sub-task is performed as follows.

- For its input wires $w \in W_{\mathsf{in}_i}$, party P_i computes $\Lambda_w = \rho_w \oplus \lambda_w$, and broadcasts the public value Λ_w to all parties.
- Each party P_j call the preprocessing functionality to open the committed keys $\boldsymbol{k}^i_{w,\Lambda_w}$, for all wires $w \in \{W_{\mathsf{in}_i}\}_{i\in[n]}$, corresponding to the public values Λ_w.
- Passing through the circuit topologically, the parties can now locally compute the following operations for each gate g. Let the gates input wires be labelled u and v, and the output wire be labelled w. Let Λ_a and Λ_b be the respective public values on the input wires.
  - Each party computes, for all $j \in [n]$:

$$k^j_{w,\Lambda_c} = \tilde{g}^j_{\Lambda_a,\Lambda_b} \oplus \left(\bigoplus_{i=1}^{n} F_{\boldsymbol{k}^i_{u,\Lambda_a}}(g\|\Lambda_b\|j) \bigoplus_{i=1}^{n} F_{\boldsymbol{k}^i_{v,\Lambda_b}}(g\|\Lambda_a\|j) \right)$$

    Otherwise, it proceeds.
  - If $\boldsymbol{k}^i_{w,\Lambda_c} = \boldsymbol{k}^i_{w,0}$ then P_i sets $\Lambda_c = 0$; if $\boldsymbol{k}^i_{w,\Lambda_c} = \boldsymbol{k}^i_{w,1}$ then P_i sets $\Lambda_c = 1$.
  1. The output of the gate is defined to be $(\boldsymbol{k}^1_{w,\Lambda_c}, \ldots, \boldsymbol{k}^n_{w,c})$ and the public value Λ_c.
- At the end of the circuit evaluation, everyone obtains a public value Λ_w, for all $w \in W_{\mathsf{out}}$. The parties can then recover the actual outputs from $y_w = \Lambda_w \oplus \lambda_w$, where λ_w was obtained in the preprocessing stage.

Fig. 8: Robust BMR protocol

their masked input and the response containing the keys corresponding to the masked values. After this, each party computes the output by locally un-garbling the tables from the preprocessing with the keys that they received. Assuming that each party behaves honestly, this online phase fails to be correct only if the un-garbling does.

However, the un-garbling only consists in removing deterministic pseudorandom masks from the garbled values which are produced using the point-and-permute technique [PSSW09]. Assuming that parties behave honestly in the broadcast and in the re-computation of these masks, this will always be perfectly correct. Indeed, given public input values $\Lambda_u = \rho_u \oplus \lambda_u, \Lambda_v = \rho_v \oplus \lambda_v$ corresponding to the input wires u, v, parties will be able to "decrypt" a single ciphertext and obtain the keys

$$\boldsymbol{k}^j_{w,\Lambda_w} = \boldsymbol{k}_{w,0} \oplus (g(\rho_u, \rho_v) \oplus \lambda_w) \cdot (\boldsymbol{k}^j_{w,0} \oplus \boldsymbol{k}^j_{w,1}),\ \forall j,$$

where $\Lambda_w = g(\rho_u, \rho_v) \oplus \lambda_w$, for each gate g. This therefore implies that the protocol $\Pi^{\mathtt{off}}_{\mathsf{RobBMR}}$ has perfect online correctness.

We now prove security of our protocol. Let $\mathcal{A}$ be a PPT adversary corrupting a subset of parties $A \subset [n]$ such that $|A| = t_p$. We describe a PPT simulator $\mathcal{S}$, with access to an ideal functionality $\mathcal{F}$ that implements f, which simulates the adversary's view. A key $\boldsymbol{k}_w$ for wire w is denoted as an active key if it is observed by the adversary upon evaluating the garbled circuit. The remaining

hidden key is denoted as an inactive key. An active path is the set of all active keys that are observed throughout the garbled circuit evaluation.

Denoting the set of honest parties by H, our simulator $\mathcal{S}$ is defined below.

The description of the simulation.

1. Initialization. Upon receiving the adversary's input $(1^\kappa, A, \boldsymbol{x}_A)$ and output $\boldsymbol{y}$, $\mathcal{S}$ samples a i.i.d uniformly random tapes r_i for each $i \in A$, incorporates $\mathcal{A}$ and internally emulates an execution of the honest parties running $\Pi^{\mathsf{on}}_{\mathsf{RobBMR}}$ with the adversary $\mathcal{A}$. When we say that $\mathcal{S}$ chooses a value for some corrupted party, we mean that it samples the value from that party's random tape r_i.
2. Preprocessing. $\mathcal{S}$ obtains the adversary's input C_f which is a Boolean circuit that computes f with a set of wires W and a set of G gates, and emulates $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$, as follows:
   - For every input wire $w \in W_{\mathsf{in_i}}$ the simulator chooses a random bit $\Lambda_w \in \{0,1\}$ and, for every $i \in H$, an active key $\boldsymbol{k}^i_{w,\Lambda_w}$. Additionally, it chooses a key $\boldsymbol{k}^i_{w,1-\Lambda_w} \in \{0,1\}^\kappa$, for every $i \in A$. Generate the relative commitments for the input-wire keys, sampling random values for the inactive keys of parties in H.
   - For every $w \in \{W_{\mathsf{in_i}}\}_{i\in[n]}$, $\mathcal{S}$ samples keys $\{\boldsymbol{k}^i_{w,0}, \boldsymbol{k}^i_{w,1}\}_{i\in A} \in \{0,1\}^{2\kappa}$ at random and chooses a random $\Lambda_w \in \{0,1\}$

   The simulator continues the emulation of the garbling phase by computing an active path of the garbled circuit that corresponds to the sequence of keys which will be observed by the adversary. Importantly, $\mathcal{S}$ never samples the inactive keys $k^i_{u,\bar{\Lambda}_u}, k^i_{v,\bar{\Lambda}_v}$ and $k^i_{w,\bar{\Lambda}_w}$ for $i \in \bar{A}$ in order to generate the garbled circuit.
3. Active path generation of logical gates. For every gate g with input wires $I = \{u, v\}$ and an output wire w, $\mathcal{S}$ samples a random $\Lambda_w \in \{0,1\}$ and honestly generates the entry in row (Λ_u, Λ_v), where Λ_u (resp. Λ_v) is the public value associated to the left (resp. right) input wire to g. Namely, the simulator computes

$$\langle \tilde{g}^j_{\Lambda_w,\Lambda_w} \rangle = \langle \left(\bigoplus_{i=1}^{n} F_{k^i_{u,\Lambda_u}}(g \| \Lambda_v, \| j) \oplus F_{k^i_{v,\Lambda_v}}(g \| \Lambda_u \| j) \right) \oplus k^j_{w,\Lambda_w} \rangle.$$

   The remaining three rows are sampled uniformly at random from $\{0,1\}^\kappa$.
4. Setting the translation table. For every output wire $w \in W_{\mathsf{out}}$ returning the ith bit of $\boldsymbol{y}$, the simulator sets $\lambda_w = \Lambda_w \oplus y_i$. For all input wires $w \in W_{\mathsf{in_i}}$ that are associated with the ith bit of $\boldsymbol{x}_A$ (the adversary's input), the simulator sets $\lambda_w = \Lambda_w \oplus \boldsymbol{x}_{A,i}$. The simulator forwards the adversary the λ_w value for every output wire $w \in W_{\mathsf{out}}$ and every circuit-input wire $w \in W_{\mathsf{in_i}}$ associated with a corrupted party. It completes the emulation of $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$ by adding the complete garbled circuit to the view of each corrupted party.
5. Online computation. In the online computation the simulator adds to the view of every corrupted party the public values $\{\Lambda_w\}_{w\in W_{\mathsf{in_i}}}$ that are associated with the honest parties' input wires $W_{\mathsf{in_i}}$. The simulator adds the honest parties' input keys $\{\boldsymbol{k}^i_{w,\Lambda_w}\}_{i\in\bar{A}, w\in W_{\mathsf{in_i}}}$ and corresponding decommitment values to the view of each corrupted party.

This concludes the description of the simulation. Note that the difference between the simulated and the real executions is regarding the way the garbled circuit is generated. More concretely, the simulated garbled gates include a single row that is properly produced, whereas the remaining

three rows are picked at random. Let $\mathbf{HYB}^{\mathcal{F}_{\mathsf{RobBMR}}^{\mathsf{off}}}_{\Pi_{\mathsf{BMR}},\mathcal{A},\mathcal{Z}}(1^\kappa, z)$ denote the output distribution of the adversary $\mathcal{A}$ and honest parties in a real execution using Π_{BMR} with adversary $\mathcal{A}$. Moreover, let $\mathbf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(1^\kappa, z)$ denote the output distribution of $\mathcal{S}$ and the honest parties in an ideal execution. We prove that the ideal and real executions are indistinguishable.

Lemma 4. *The following two distributions are computationally indistinguishable:*

- $\{\mathbf{HYB}^{\mathcal{F}_{\mathsf{RobBMR}}^{\mathsf{off}}}_{\Pi_{\mathsf{BMR}},\mathcal{A},\mathcal{Z}}(1^\kappa, z)\}_{\kappa\in\mathbb{N},z\in\{0,1\}^*}$
- $\{\mathbf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(1^\kappa, z)\}_{\kappa\in\mathbb{N},z\in\{0,1\}^*}$

Proof. We begin by defining a slightly modified simulated execution $\widetilde{\mathbf{HYB}}$, where the generation of the garbled circuit is modified so that upon receiving the parties' inputs $\{\rho^i\}_{i\in[n]}$ the simulator $\mathcal{S}'$ first evaluates the circuit C_f, computing the actual bit ρ_w to be transferred via wire w for all $w \in W$, where W is the set of wires of C_f. It then chooses wire mask shares and wire keys as $\mathcal{F}_{\mathsf{RobBMR}}^{\mathsf{off}}$. Finally, $\mathcal{S}'$ fixes the active key for each wire $w \in W$ to be $(k^1_{w,\rho_w\oplus\lambda_w}, \ldots, k^n_{w,\rho_w\oplus\lambda_w})$. The rest of this hybrid is identical to the simulation. This hybrid execution is needed in order to construct a distinguisher for the PRF.
For completeness, we recall here the notion of pseudo-random function under multiple keys.

Definition 17 (PRF under multiple keys). *Let $F : \{0,1\}^\kappa \times [|G|] \times \{0,1\} \times [n] \to \{0,1\}^\kappa$ be an efficient, length preserving, keyed function. F is a* pseudo-random function under multiple keys *if for all PT distinguisher $\mathcal{D}$, there exists a negligible function* negl *such that:*

$$|\Pr[\mathcal{D}^{F_{\bar{k}}(\cdot)}(1^\kappa) = 1] - \Pr[\mathcal{D}^{\bar{f}(\cdot)}(1^\kappa) = 1] \leq \mathsf{negl}(\kappa),$$

where $F_{\bar{k}} = F_{k_1}, \ldots, F_{k_{m(n)}}$ are pseudorandom function F keyed with polynomial number of randomly chosen keys $k_1, \ldots, k_{m(n)}$ and $\bar{f} = f_1, \ldots, f_{m(n)}$ are $m(n)$ random functions from $\{0,1\}^n$ mapto $\{0,1\}^n$. The probability in both cases is taken over the randomness of $\mathcal{D}$.

Let $\widetilde{\mathbf{HYB}}^{\mathcal{F}_{\mathsf{RobBMR}}^{\mathsf{off}}}_{\Pi_{\mathsf{BMR}},\mathcal{A}}(1^\kappa, z)$ denote the output distribution of the adversary $\mathcal{A}$ and honest parties in this game. It is simple to verify that the adversary's views in $\widetilde{\mathbf{HYB}}$ and $\mathbf{IDEAL}$ are identical, as in both cases the garbling of each gate includes just a single row that is correctly garbled.

Assume by contradiction the existence of an environment $\mathcal{Z}$, an adversary $\mathcal{A}$ and a non-negligible function $p(\cdot)$ such that

$$\left|\Pr[\mathcal{Z}(\mathbf{HYB}^{\mathcal{F}_{\mathsf{RobBMR}}^{\mathsf{off}}}_{\Pi_{\mathsf{BMR}},\mathcal{A},\mathcal{Z}}(1^\kappa, z)) = 1] - \Pr[\mathcal{Z}(\widetilde{\mathbf{HYB}}_{\Pi_{\mathsf{BMR}},\mathcal{A},\mathcal{Z}}(1^\kappa, z)) = 1]\right| \geq \frac{1}{p(\kappa)}$$

for infinitely many κ's where the probability is taken over the randomness of $\mathcal{Z}$ as well as the randomness for choosing the Λ values and the keys. Then we construct a PPT distinguisher $\mathcal{D}$ for PRF that distinguishes between an instance of the form

$$\left(F, \bigoplus_{i\in H} F_{\boldsymbol{k}^i}(g\|0\|j), \bigoplus_{i\in H} F_{\bar{\boldsymbol{k}}^i}(g\|0\|j), \bigoplus_{i\in H} F_{\boldsymbol{k}^i}(g\|1\|j), \bigoplus_{i\in H} F_{\bar{\boldsymbol{k}}^i}(g\|1\|j)\right)$$

and five random elements, for some subset H of $[n]$ of size $n - t_p$ (that corresponds to the set of honest parties) with probability at least $\frac{1}{p(\kappa)\cdot|C|}$ via a sequence of hybrid games $\{\mathbf{HYB}_\ell\}_{\ell\in[|C|]}$,

where $|C|$ = number of logical gates in C. In more details, we define hybrid $\mathbf{HYB}_\ell$ as a hybrid execution with a simulator $\mathcal{S}_\ell$ that garbles the circuit as follows. The first ℓ gates in the topological order are garbled as in the simulation whereas the remaining $|C|-\ell$ gates are garbled as in the real execution. Note that $\mathbf{HYB}_0$ is distributed as hybrid $\mathbf{HYB}$ and that $\mathbf{HYB}_{|C|}$ is distributed as $\widetilde{\mathbf{HYB}}$. Therefore, if $\mathbf{HYB}$ and $\widetilde{\mathbf{HYB}}$ are distinguishable with probability $\frac{1}{p(\kappa)}$ then there exists $\tau \in [|C|]$ such that hybrids $\mathbf{HYB}_{\tau-1}$ and $\mathbf{HYB}_\tau$ are distinguishable with probability at least $\frac{1}{p(\kappa)\cdot|C|}$. Next, we formally describe our reduction to PRF security. Upon receiving a tuple $(\tilde{F}, \tilde{F}_0, \tilde{F}'_0, \tilde{F}_1, \tilde{F}'_1)$ that is distributed according to the first or the second distribution, a subset H of $[n]$ that denotes the set of honest parties, an index τ and the environment's input z, distinguisher $\mathcal{D}$ internally invokes $\mathcal{Z}$ and simulator $\mathcal{S}$. In more details,

- $\mathcal{D}$ internally invokes $\mathcal{Z}$ that fixes the honest parties' inputs ρ.
- $\mathcal{D}$ emulates the communication with the adversary (controlled by $\mathcal{Z}$) in the initialization, pre-processing and garbling steps as in the simulation with $\mathcal{S}$.
- For each wire u, let $\rho_u \in \{0,1\}$ be the actual value on wire u. Note that these values, as well as the output of the computation y, can be determined since $\mathcal{D}$ knows the actual input of all parties to the circuit.
- For each wire u in the circuit and $i \in A$, $\mathcal{D}$ chooses a pair of keys $\boldsymbol{k}^i_{u,0}, \boldsymbol{k}^i_{u,1} \in \{0,1\}^\kappa$, whereas for all $i \in H$ it samples a random key $\boldsymbol{k}^i_{u,\Lambda_u} \in \{0,1\}^\kappa$. $\mathcal{D}$ further fixes the public value $\Lambda_u = \lambda_u \oplus \rho_u$.
- $\mathcal{D}$ then garbles the circuit as follows.
  - For every g_ι with input wires u and v and output wire w, $\mathcal{D}$ continues as follows.
    If $\iota < \tau$ then $\mathcal{D}$ garbles g_j exactly as in the simulation with $\mathcal{S}'$.
    If $\iota = \tau$ then $\mathcal{D}$ first honestly computes the (Λ_u, Λ_v)-th row by fixing

$$\tilde{g}^j_{\Lambda_u,\Lambda_v} = \left(\bigoplus_{i=1}^{n} F_{\boldsymbol{k}_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\boldsymbol{k}_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right) \oplus \boldsymbol{k}^j_{w,\Lambda_w}.$$

    Next, $\mathcal{D}$ samples an inactive key $k^i_{w,\bar{\Lambda}_w}$ for all $i \in \bar{A}$ and fixes the remaining three rows as follows.

$$\tilde{g}^j_{\Lambda_u,\bar{\Lambda}_v} = \left(\bigoplus_{i=1}^{n} F_{\boldsymbol{k}^i_{u,\Lambda_u}}(g\|\bar{\Lambda}_v\|j) \oplus \left(\bigoplus_{i\in A} F_{\boldsymbol{k}_{v,\bar{\Lambda}_v}}(g\|\Lambda_u\|j)\right) \oplus \tilde{F}'_{\Lambda_u}\right) \oplus \boldsymbol{k}^j_{w,c},$$
$$\text{where} \quad c = \Lambda_u \cdot \bar{\Lambda}_v \oplus \Lambda_w \oplus \rho_w$$
$$\tilde{g}^j_{\bar{\Lambda}_u,\Lambda_v} = \left(\bigoplus_{i\in A} F_{\boldsymbol{k}_{u,\bar{\Lambda}_u}}(g\|\Lambda_v\|j) \oplus \tilde{F}_{\Lambda_v} \oplus \left(\bigoplus_{i=1}^{n} F_{\boldsymbol{k}_{v,\Lambda_v}}(g\|\bar{\Lambda}_u\|j)\right)\right) \oplus \boldsymbol{k}^j_{w,c},$$
$$\text{where} \quad c = \bar{\Lambda}_u \cdot \Lambda_v \oplus \Lambda_w \oplus \rho_w$$
$$\tilde{g}^j_{\bar{\Lambda}_u,\bar{\Lambda}_v} = \left(\bigoplus_{i\in A} F_{\boldsymbol{k}_{u,\bar{\Lambda}_u}}(g\|\bar{\Lambda}_v\|j) \oplus \tilde{F}_{\bar{\Lambda}_v} \oplus \left(\bigoplus_{i\in A} F_{\boldsymbol{k}_{v,\bar{\Lambda}_v}}(g\|\bar{\Lambda}_u\|j)\right) \oplus \tilde{F}'_{\bar{\Lambda}_u}\right)$$
$$\oplus \boldsymbol{k}^j_{w,c}, \quad \text{where} \quad c = \bar{\Lambda}_u \cdot \bar{\Lambda}_v \oplus \Lambda_w \oplus \rho_w.$$

    Finally, if $\iota > \tau$ then $\mathcal{D}$ garbles g_ι exactly as in hybrid $\mathbf{HYB}$. For that, $\mathcal{D}$ needs to know both active and inactive keys. It therefore chooses the inactive keys that are associated with the input and output wires of this gate for $i \in \bar{A}$, in order to be able to complete the garbling. Recall that the circuit is with fan-out 1. Therefore the distinguisher can choose the inactive key for the input wire of this gate (as it was not used as an input wire to gate g_τ).

– This concludes the description of the reduction. $\mathcal{D}$ hands the adversary the complete description of the garbled circuit and concludes the execution as in the simulation with $\mathcal{S}'$.
– $\mathcal{D}$ outputs whatever $\mathcal{Z}$ does.

Note first that if $(\tilde{F}, \tilde{F}_0, \tilde{F}'_0, \tilde{F}_1, \tilde{F}'_1)$ are truly uniform then the view generated by $\mathcal{D}$ is distributed as in $\mathbf{HYB}_\tau$. This is because only the active path is created as in the real execution, whereas the remaining rows are sampled uniformly at random from the appropriate domain. On the other hand, if this tuple is generated according to the following distribution

$$\left(F, \bigoplus_{i\in H} F_{\boldsymbol{k}^i}(g\|0\|j), \bigoplus_{i\in H} F_{\bar{\boldsymbol{k}}^i}(g\|0\|j), \bigoplus_{i\in H} F_{\boldsymbol{k}^i}(g\|1\|j), \bigoplus_{i\in H} F_{\bar{\boldsymbol{k}}^i}(g\|1\|j)\right)$$

then this emulates game $\mathbf{HYB}_{\tau-1}$, since each tuple element emulates an evaluation of the hash values for the honest parties on the secret keys.

This completes the proof of the lemma and proposition

□

Robustness. Let $\mathtt{out}_1^{\mathtt{off}}, \ldots, \mathtt{out}_n^{\mathtt{off}}$ be the output provided by the ideal functionality $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$. Assume that exists a party, say $\mathtt{P}_j$, in an execution of $\Pi^{\mathtt{on}}_{\mathsf{RobBMR}}$ which outputs $\mathtt{out}^j \neq y$, where $y = f(x, w_1, \ldots, w_n)$ with (x, w_i) be the input of party $\mathtt{P}_i$, for each $i \in [n]$.

For this to happen one of the following two events must occur:

1. The circuit must be incorrectly garbled so as to output an incorrect result;
2. The corrupted parties provide keys during the second broadcast such that these flip the output bits of certain gates within the garbled circuit.

By assumption, event 1 does not happen since the output of $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$ is assumed to be correct. It is therefore event 2 that must occur.

Assume that the adversary controls party $\mathtt{P}_1$ and its keys $\boldsymbol{k}^1_{w,\Lambda_w}$ for input wires $w \in W_{\mathsf{in}j}$, for $j \in [n]$. We recall that each honest party $i \neq 1$ will compute the following for every $j \in [n]$, and gate g in the circuit:

$$\tilde{g}^j_{\Lambda_u,\Lambda_v} \oplus \left(\bigoplus_{i=1}^{n} \left(F_{\boldsymbol{k}^i_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\boldsymbol{k}^i_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right)\right). \tag{2}$$

From this tuple, each party $\mathtt{P}_i$ then compares its attributed value $\boldsymbol{k}^i_{w,\Lambda_c}$ with $\{\boldsymbol{k}^i_{w,0}, \boldsymbol{k}^i_{w,1}\}$ (contained in $\mathtt{out}_i^{\mathtt{off}}$) to set the public output value Λ_w. This is where a bit flip could happen. Assuming that the offline phase is correct, in particular the gates are correctly garbled, the only point where the adversary can cheat is when the parties broadcast the keys $\boldsymbol{k}^i_{w,\Lambda_w}$ corresponding to the masked input Λ_w. Here the adversary can broadcast any value $\tilde{\boldsymbol{k}}^1_{w,\Lambda_w}$ (remember we are assuming $\mathtt{P}_1$ corrupt).

During the evaluation of one of the subsequent gates, the corrupt party has to manage to flip the value Λ_w for (at least) one honest party. This implies that the two input keys provided to $\mathtt{P}_1$ have to create a flip for some $\mathtt{P}_j$. In Equation (2), this means that:

$$\begin{aligned}\tilde{g}^j_{\Lambda_u,\Lambda_v} \oplus \left(\bigoplus_{i=2}^{n} \left(F_{\boldsymbol{k}^i_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\boldsymbol{k}^i_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right)\right) \\ \oplus \left(F_{\tilde{\boldsymbol{k}}^1_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\tilde{\boldsymbol{k}}^1_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right) = \boldsymbol{k}^j_{w,1-\Lambda_w}.\end{aligned}$$

Recall that

$$\tilde{g}^{j}_{\Lambda_u,\Lambda_v} = \left(\bigoplus_{i=1}^{n}\left(F_{\boldsymbol{k}^{i}_{u,\Lambda_u}}(g\|\Lambda_u\|j) \oplus F_{\boldsymbol{k}^{i}_{v,\Lambda_v}}(g\|\Lambda_v\|j)\right)\right) \oplus \boldsymbol{k}^{j}_{w,\Lambda_w},$$

so to successfully flip all the key corresponding to $\mathtt{P}_j$ we should have:

$$\begin{aligned}&\left(F_{\tilde{\boldsymbol{k}}^{1}_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\tilde{\boldsymbol{k}}^{1}_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right) = \\ &\left(F_{\boldsymbol{k}^{1}_{u,\Lambda_u}}(g\|\Lambda_v\|j) \oplus F_{\boldsymbol{k}^{1}_{v,\Lambda_v}}(g\|\Lambda_u\|j)\right) \oplus (\boldsymbol{k}^{j}_{\Lambda_w} \oplus \boldsymbol{k}^{j}_{1\oplus\Lambda_w}),\ \text{for some } j \in [n].\end{aligned}$$

This means that an unbounded adversary can break the robustness of the protocol, unless it is committed to use the input-wire keys provided by $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$. For this reason we make sure, during the preprocessing protocol, that each party commits to the input-wire keys, i.e. in the preprocessing each $\mathtt{P}_i$ provides $\mathsf{Com}(i, w, 0, \boldsymbol{k}^{i}_{w,0})$ and $\mathsf{Com}(i, w, 1, \boldsymbol{k}^{i}_{w,1})$, where Com is a statistically biding commitment [Nao90]. Since the pre-processing is correct, this is enough to guarantee robustness.. More formally, we can prove the following.

Proposition 2. *The protocol Π_{RobBMR} of Figure 8 is robust (in the sense of Definition 16).*

Proof. Robustness requires that $\Pi^{\mathtt{off}}_{\mathsf{RobBMR}}$ is honestly executed, in particular that all the keys and the garbled circuit $\tilde{C}_f$ are correct and consistent, i.e. exists an evaluation procedure $\mathcal{B}$ such that $\mathcal{B}(\tilde{C}_f(\boldsymbol{x}, R)) = C_f(\boldsymbol{x})$, where C_f is the circuit computing f. The fact that Π_{RobBMR} is robust follows from the fact that given the material generated in $\mathcal{F}^{\mathtt{off}}_{\mathsf{RobBMR}}$ and the input step, which fixes the input and corresponding keys, the online views generated by the parties are identical and cannot lead to an incorrect result, i.e. the view generated by each $\mathtt{P}_i$ results in the unique active path that corresponds to the evaluation of the garbled circuit which is deterministic. The only attempt an adversary can make to break robustness is by broadcasting incorrect input keys $\boldsymbol{k}_{w,\Lambda}, w \in W_{\mathsf{in}}$, therefore breaking the binding property of the commitment scheme.

□

Acknowledgements. We thank Carmit Hazay and Muthuramakrishnan Venkitasubramaniam for insightful discussions on the MPC-in-the-head approach. Emmanuela Orsini was supported by the Defense Advanced Research Projects Agency (DARPA) under contract No. HR001120C0085, and by CyberSecurity Research Flanders with reference number VR20192203. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA, the US Government or Cyber Security Research Flanders. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Bar02. Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In FOCS. 2002.

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, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 459–487, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

BMR90. Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513, Baltimore, MD, USA, May 14–16, 1990. ACM Press.

CCG+20. Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 291–319, Durham, NC, USA, November 16–19, 2020. Springer, Heidelberg, Germany.

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 383–395, Portland, Oregon, October 21–23, 1985. IEEE Computer Society Press.

CLP20. Rohit Chatterjee, Xiao Liang, and Omkant Pandey. Improved black-box constructions of composable secure computation. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *ICALP 2020: 47th International Colloquium on Automata, Languages and Programming*, volume 168 of *LIPIcs*, pages 28:1–28:20, Saarbrücken, Germany, July 8–11, 2020. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

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, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 270–299, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

COSV17a. Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four-round concurrent non-malleable commitments from one-way functions. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 127–157, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

COSV17b. Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Round-optimal secure two-party computation from trapdoor permutations. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 678–710, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

CPS+16. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Improved OR-composition of sigma-protocols. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 112–141, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

CPV20. Michele Ciampi, Roberto Parisella, and Daniele Venturi. On adaptive security of delayed-input sigma protocols and fiat-shamir NIZKs. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20: 12th International Conference on Security in Communication Networks*, volume 12238 of *Lecture Notes in Computer Science*, pages 670–690, Amalfi, Italy, September 14–16, 2020. Springer, Heidelberg, Germany.

CRSW22. Michele Ciampi, Divya Ravi, Luisa Siniscalchi, and Hendrik Waldner. Round-optimal multi-party computation with identifiable abort. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 335–364. Springer, 2022.

CVZ10. Zhenfu Cao, Ivan Visconti, and Zongyang Zhang. Constant-round concurrent non-malleable statistically binding commitments and decommitments. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2010.

DDN91. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, LA, USA, May 6–8, 1991. ACM Press.

DMRV13. Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Muthuramakrishnan Venkitasubramaniam. Adaptive and concurrent secure computation from new adaptive, non-malleable commitments. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 316–336, 2013.

FLS90. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science*, pages 308–317, St. Louis, MO, USA, October 22–24, 1990. IEEE Computer Society Press.

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

GK96. Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

GKS16. Vipul Goyal, Dakshita Khurana, and Amit Sahai. Breaking the three round barrier for non-malleable commitments. In *57th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2016*. IEEE, 2016.

GL89. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989.

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

Gol06. Oded Goldreich. *Foundations of Cryptography: Volume 1*. Cambridge University Press, New York, NY, USA, 2006.

Goy11. Vipul Goyal. Constant round non-malleable protocols using one way functions. In STOC. 2011.

GR19. Vipul Goyal and Silas Richelson. Non-malleable commitments using Goldreich-Levin list decoding. In David Zuckerman, editor, *60th Annual Symposium on Foundations of Computer Science*, pages 686–699, Baltimore, MD, USA, November 9–12, 2019. IEEE Computer Society Press.

GRRV14a. Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *55th Annual Symposium on Foundations of Computer Science*, pages 41–50, Philadelphia, PA, USA, October 18–21, 2014. IEEE Computer Society Press.

GRRV14b. Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. Cryptology ePrint Archive, Paper 2014/586, 2014. `https://eprint.iacr.org/2014/586`.

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, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 488–520, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

HV16. Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 397–429, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th Annual ACM Symposium on Theory of Computing*, pages 21–30, San Diego, CA, USA, June 11–13, 2007. ACM Press.

Khu17. Dakshita Khurana. Round optimal concurrent non-malleability from polynomial hardness. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 139–171, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

KOS18. Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. Round optimal black-box "commit-and-prove". In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 286–313, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

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.

LP15. Huijia Lin and Rafael Pass. Constant-round nonmalleable commitments from any one-way function. *J. ACM*, 62(1):5:1–5:30, 2015.

LPV08. Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In TCC. 2008.

MP12. Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 701–718. Springer, 2012.

Nao90. Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany.

OPV09. Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Simulation-based concurrent non-malleable commitments and decommitments. In Omer Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 91–108. Springer, 2009.

Pas13. Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In *TCC*, pages 334–354, 2013.

PR03. Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 404–413. IEEE Computer Society, 2003.

PR05a. Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 563–572, 2005.

PR05b. Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In STOC. 2005.

PR08a. Rafael Pass and Alon Rosen. Concurrent nonmalleable commitments. *SIAM J. Comput.*, 37(6):1891–1925, 2008.

PR08b. Rafael Pass and Alon Rosen. New and improved constructions of nonmalleable cryptographic protocols. *SIAM J. Comput.*, 38(2):702–752, 2008.

PSSW09. Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 250–267, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.

PW10. Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitments from sub-exponential one-way functions. In EUROCRYPT. 2010.

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

Yao82. Andrew Chi-Chih Yao. Space-time tradeoff for answering range queries (extended abstract). In *14th Annual ACM Symposium on Theory of Computing*, pages 128–136, San Francisco, CA, USA, May 5–7, 1982. ACM Press.