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

Michele Ciampi[1][0000−0001−5062−0388], Emmanuela Orsini[2][0000−0002−1917−1833], and Luisa Siniscalchi[3,4][0000−0003−1813−1132]

[1] The University of Edinburgh, Edinburgh, UK
[2] imec-COSIC, KU Leuven, Leuven, Belgium.
[3] Dept. Computer Science, Aarhus University, Aarhus, Denmark.
[4] Concordium Blockchain Research Center, 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 any (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. [15], 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 [44, 3], 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 [10, 26, 1, 7].

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) [19] 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 [34]. Recently many progress have been made also for the case of *non-malleable* (NM) commitments[5]. Indeed, the long sequence of very exciting positive re-

[5] In this paper we will consider only NM commitments w.r.t. commitments. For the case of NM w.r.t. decommitments see [39, 41, 35, 4, 14, 21].

sults [2, 37, 39, 38, 41, 40, 33, 42, 43, 31, 32, 20, 22, 24, 9, 8] led to the work of Khurana [29] in which the authors showed how to obtain a 3-round (which is optimal for the case of polynomial-time assumptions [36]) non-malleable commitment scheme based on specific number-theoretic assumptions, and to [23] 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 [43, 31, 20, 22], but their round complexity is far to be optimal. More specifically, Goyal et al. [22] 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. [24] mention that combining their protocol with ideas from [22] would could to a 6-round protocol but no explicit construction 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.[6] 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* [27, 12, 11]. These properties guarantee that HVZK and soundness hold even against an adversary that decides the statement to be

[6] 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.

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 [27]. 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 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 [7, 23, 13]. And, more interestingly, it was used also to construct the first one-one non-malleable commitment [24][7]. As we will discuss in the technical overview, we will replace the rewind secure proof system proposed in [24] (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. To the best of our knowledge, no black-box rewind secure three-round HVZK protocol is currently available. 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.*

[7] In Section 8 we propose a comparison between the approach based on rewind-secure primitives of [24] 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 [24].

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 [28]. 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.[8]

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 [28] 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

[8] 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.

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[9]. Compared to a standard commitment scheme, they can be opened in two modes: 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.

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.

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

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. [22], 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[10]. 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 [30]. 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 [30], 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

[10] 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.

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 [3] is 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 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 full version, we 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. [24]. 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). To lift the security of such a commitment and build a fully non-malleable commitment scheme, [24] run, in parallel to Π_{wnmc}, a zero-knowledge proof.

As noted in [24, 9], 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 [24] 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 [23, 24], which, unfortunately, make non-black-box use of the underlying primitives. In [9], 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. 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 [16] (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 [30], 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

Notation. 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 than 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 [18] for more details.

Ambiguous and extractable commitments. We formally introduce the notion of *ambiguous commitments*. Compared to regular commitment schemes, with standard commitment and opening algorithms $(\mathsf{Com}, \mathsf{Dec})$, ambiguous commitments have two additional algorithms $\mathsf{Com}^{\mathsf{eq}}$ and Eq, which allow the committer to equivocate, i.e., $\mathsf{Com}^{\mathsf{eq}}$ produces an "equivocable commitment" that Eq can open to any message $m \in \{0,1\}^{\ell}$. This type of commitment schemes are sometimes called *trapdoor* or *equivocal* commitments. We provide a formal definition and construction in the full version. In this work, we also use the notion of *extractable commitments* (we refer to the full version for the formal definition). Informally, a commitment scheme is said to be extractable if there exists an efficient extractor that, having black-box access to a malicious committer that successfully performs the commitment phase, is able to extract the committed message.

3.2 Non-Malleable Commitments

Here we follow the same notation of Goyal et al. [24]. 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 1 (Non-Malleable (NM) commitment scheme [24]). *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.

3.3 Σ-Commitments

We introduce the notion of Σ-commitments, which is reminiscent of the notion of Σ-protocols.

Definition 2. *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). *There exists a polynomial-time simulator* Sim *such that for any message $m \in \{0,1\}^{\ell}$ and 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.*

3.4 Adaptive-Input SHVZK

Definition 3 (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'.*

3.5 One-of-Two Binding Commitments

We propose a formal definition of the *one-of-two binding commitments* proposed by Khurana et al. in [30]. 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, in this, 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 [30] 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 4 (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*

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

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'.*

3.6 MPC Definitions

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 $i \in [n]$.

We assume familiarity with the standard definition of MPC (referring the reader to the full version for a formal discussion), and here we formally introduce a new special property for an MPC protocol $\Pi = \Pi^{\mathtt{off},\mathtt{on}} = (\mathtt{P}_1, \ldots, \mathtt{P}_n)$.

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 [28] 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 5 (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 the full-version we show a concrete instantiation of a robust MPC protocol.

3.7 Verifiable Secret Sharing (VSS)

A verifiable secret sharing (VSS) scheme [6] 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 6 (Verifiable Secret Sharing [6, 5]). *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.

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. *The computationally unbounded adversary can corrupt up to t parties that can deviate from the above procedures. 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 [17].

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 7 (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.

5 Our Delayed-Input MPC-in-the-Head Protocol Π_{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 5).
- An ambiguous commitment scheme $\Pi_{\mathtt{com}} = (\mathsf{Com}, \mathsf{Dec}, \mathsf{Com}^{\mathsf{eq}}, \mathsf{Eq})$.

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 1. 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 $i \in I$.

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 $i_j \in$

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

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. To simplify the composition of our protocol with other primitives, we design the prover so that it expects to receive a (random) n-out-of-n secret sharing of the witness (instead of the witness itself). This is without loss of generality. 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 1. *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, then $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ (Figure 1) for the $\mathcal{NP}$-relation Rel is a 3-round public-coin delayed-input protocol satisfying adaptive-input SHVZK adaptive-input soundness with constant soundness error.*

We establish adaptive correctness, adaptive-input soundness and adaptive-input SHVZK. 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 1. *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 Π_{ComExt} against synchronizing adversaries.*

The proof of the lemma can be found in the full version.

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 [28], 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.*

6 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.

6.1 Commitment from Verifiable Secret Sharing

We start by recalling some of the techniques introduced by Goyal et al. [22]. We show that these techniques can be used to build a Σ-commitment (Definition 2) that we denote by $\Pi = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ and formally describe it in Figure 2. The protocol makes use of the following primitives:

- An $(n+1, t)$-VSS protocol $\Pi^{\mathsf{vss}} = (\Pi_{\mathsf{Share}}, \Pi_{\mathsf{Recon}})$ as defined in Definition 6. 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. [17]
- 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.

In the full version, we prove the following theorem that we shall use in the next sections.

Theorem 2. *Let* Π^{vss} *be a* $(n+1, t)$*-VSS protocol satisfying Definition 6, 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 2) is a* Σ*-commitment.*

6.2 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. Our protocol makes black-box use of the underlying primitives.

The protocol $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ is fully described in Figure 3. It makes use of the following tools:

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 $(\mathtt{com}_j^{\sigma}, \mathtt{dec}_j^{\sigma}) \leftarrow \mathsf{Com}(\mathsf{view}_j^{\sigma})$ for $j \in [n]$.
3. Set
   - $\mathtt{dec}^{\sigma} = \{\mathtt{dec}_j^{\sigma}, \mathsf{view}_j^{\sigma}\}_{j \in [n]}$
   - $\pi_1^{\sigma} = (\mathtt{com}_1^{\sigma}, \ldots, \mathtt{com}_n^{\sigma})$
4. Send π_1^{σ} to $\mathcal{R}^{\Sigma}$.

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

1. Choose a random subset $I = \{i_1, \ldots, i_k\} \subset [n]$.
2. Define and send $\pi_2^{\sigma} = (i_i, \ldots, i_k)$ 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 $\{\mathtt{dec}_j^{\sigma}, \mathsf{view}_j^{\sigma}, w_j\}_{j \in I}$.
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. 2: $\Pi = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$

- The Σ-commitment $\Sigma = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ defined in Figure 2, Section 6.1.
- The adaptive-input SHVZK $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ with adaptive-input soundness for the $\mathcal{NP}$-relation

$$\mathsf{Rel}_{\mathsf{AI}} = \{(x, a, \alpha, \{\mathsf{view}_{i_j}\}_{j \in [k]}), (r, \{\mathsf{view}_i\}_{j \in [n]}) : 1 \leq i_1 < \cdots < i_k < n \wedge w = \mathsf{Recon}(\{\mathsf{view}_i\}_{j \in [n]}) \ \wedge \ \mathsf{Rel}(x, w) = 1 \ \wedge \ a = w + r\alpha\}.$$

where Recon is the reconstruction phase of an information-theoretic $(n+1, t)$-VSS protocol Π^{vss}, with $k \leq t$. We recall that to run Π_{AI} the prover needs the statement and the witness only in the third round. Moreover, the prover expects to receive the witness in a secret shared form. We recall that 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 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. We also need Π_{AI} with the same parameters n, k, t as Σ.

At a high-level $\mathcal{P}_{\mathsf{CP}}$ commits λ^2-times to the witness w running Σ (as described in Figure 2) and proving, using $\mathcal{P}_{\mathsf{AI}}$, that each committed message w satisfies the relation Rel, and moreover that the views opened in the third round of Σ contain shares of the witness w. To make sure that the same message is committed in all the executions of Σ, we use a technique proposed by Khurana et al. in [30]. Namely, in each execution of Σ, instead of committing to w, we commit to $w||r$, for some random value r, and use the protocol Π_{AI} to additionally prove that $a = w + r\alpha$, where α is chosen as part of the second round, and a is sent in the third round from the prover. As argued in [30], 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.

More formally, we prove the following result.

Theorem 3. *Let $\Pi_{\mathsf{AI}} = (\mathcal{P}_{\mathsf{AI}}, \mathcal{V}_{\mathsf{AI}})$ be a 3-round public-coin, delayed-input complete, adaptive-input SHVZK with adaptive-input soundness for the $\mathcal{NP}$-relation $\mathsf{Rel}_{\mathsf{AI}}$, and $\Sigma = ((\mathcal{S}^{\Sigma}, \mathcal{R}^{\Sigma}), \mathsf{Dec}^{\Sigma})$ (as defined in Figure 2) be a Σ-commitment, then $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ is a 3-round public-coin adaptive-input SHVZK commit-and-prove protocol for the $\mathcal{NP}$-relation Rel.*

We first give an intuition for the adaptive-SHVZK proof by describing how the simulator $(\mathsf{Sim}^0_{\mathsf{CP}}, \mathsf{Sim}^1_{\mathsf{CP}})$ works. For ease of exposition let us focus on the i-th transcript (out of λ^2) w.r.t. challenge $(\alpha, \pi_{2,i})$, where $\pi_{2,i}$ is composed by two sets of indices I, C. The simulator $\mathsf{Sim}^0_{\mathsf{CP}}$ on input challenge $\pi_{2,i}$ runs the HRH simulator of Σ on input I obtaining $\pi^{\sigma}_1, \pi^{\sigma}_3$ and, consequently, the shares $\{\mathsf{view}^{\sigma}_{i_j}\}_{i_j \in I}$ which will be opened in the third round (denoted by π^{σ}_3). $\mathsf{Sim}^0_{\mathsf{CP}}$ then runs $\mathsf{Sim}^0_{\mathsf{AI}}$ on input $\pi_{2,i}$ thus obtaining $(\pi_{1,i}, \mathsf{aux})$. The simulator $\mathsf{Sim}^1_{\mathsf{CP}}$ on input theorem x samples a at random, sets $X = \{(x, a, \alpha, \{\mathsf{view}^{\sigma}_{i_j}\}_{i_j \in I})$ and runs $\mathsf{Sim}^1_{\mathsf{AI}}$ on input theorem (X, aux) thus obtaining $\pi_{3,i}$.

The full proof of Theorem 3 can be found in the full version. Similarly to previous protocols, we have the following result.

Corollary 2. *Assuming the existence of one-to-one one-way functions, there exists a 3-round public-coin adaptive-input SHVZK commit-and-prove Π_{CP} for the $\mathcal{NP}$-relation Rel that makes black-box use of the* 1-1 *OWFs.*

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. Hence, each round of Π_{CP} consists of two distinct components (e.g., the i-th round of Π_{CP} will be denoted by $\{\pi_i, \pi^{\sigma}_i\}$).

PUBLIC INPUT AND PARAMETERS: Parameters k, n, t of Σ, with $k = t$. $\mathcal{P}_{\mathsf{CP}}$ 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 $i \in [\lambda^2]$, do the following:
   2.1. Execute $\mathcal{S}^{\Sigma}$ on input $(1^{\lambda}, w||r)$, obtaining $\pi^{\sigma}_{i,1} = \{\mathsf{com}^{\sigma}_{i,j}\}_{j\in[n]}$, $\{\mathsf{dec}^{\sigma}_{i,j}\}_{j\in[n]}$ and $\{\mathsf{view}^{\sigma}_{i,j}\}_{j\in[n]}$.
   2.2. Run $\mathcal{P}_{\mathsf{AI}}$ on input 1^{λ}, obtaining $\pi_{i,1}$.
3. Define and send $\pi_1 = \{\pi_{i,1}, \pi^{\sigma}_{i,1}\}_{i\in[\lambda^2]}$ to $\mathcal{V}_{\mathsf{CP}}$.

Round 2 $\mathcal{V}_{\mathsf{CP}}$ computes the following steps:

1. For $i \in [\lambda^2]$, run $\mathcal{V}_{\mathsf{AI}}$ to choose challenge $\pi_{i,2}$.
2. Sample $\alpha \leftarrow \mathbb{F}$.
3. Set $\pi_2 = \{\pi_{i,2}\}_{i\in[\lambda^2]}$ and send (α, π_2) to $\mathcal{P}_{\mathsf{CP}}$.

Round 3 $\mathcal{P}_{\mathsf{CP}}$ performs the following steps:

1. Compute $a = w + r\alpha$ and, for each $i \in [\lambda^2]$, do as follows.
   1.1. Parse $\pi_{i,2}$ as $(I, q_{i_1}, \ldots, q_{i_k})$.
   1.2. Compute the 3rd message $\pi^{\sigma}_{i,3}$ of Σ executing $\mathcal{S}^{\Sigma}$ on input I (note that $\pi^{\sigma}_{i,3} = \{\mathsf{dec}^{\sigma}_{i,j}, \mathsf{view}^{\sigma}_{i,j}\}_{j\in I}$).
   1.3. Run $\mathcal{P}_{\mathsf{AI}}$ on input the pair statement-witness[a] $((x, a, \alpha, \{\mathsf{view}^{\sigma}_{i,j}\}_{j\in I}), \{\mathsf{view}^{\sigma}_{i,j}\}_{j\in[n]})$ and $\pi_{i,2}$, thus obtaining the third round $\pi_{i,3}$.
   1.4. Set $\pi_3 = \{\pi^{\sigma}_{i,3}, \pi_{i,3}, a\}_{i\in[\lambda^2]}$ and send π_3 to $\mathcal{V}_{\mathsf{CP}}$.

Verification step. On input x outputs 1 if and only if, for each $i \in [\lambda^2]$, the following holds:

1. $\mathcal{R}^{\Sigma}$ accepts the commitment $(\pi^{\sigma}_{i,1}, I, \pi^{\sigma}_{i,3})$.
2. $\mathcal{V}_{\mathsf{AI}}$ accepts the proof $(\pi_{i,1}, \pi_{i,2}, \pi_{i,3})$ for the statement $(x, a, \alpha, \{\mathsf{view}^{\sigma}_{i,j}\}_{j\in I})$.
3. The j-th view $\mathsf{view}^{\sigma}_{i,j}$ that appears in $\pi^{\sigma}_{i,3}$ represents also the input share for the j-th view $\mathsf{view}_{i,j}$ that appears in $\pi_{i,3}$, for each $j \in I$.

Decommitment procedure: On input an accepting transcript of the protocol, and on input all the decommitment information for the λ^2 commitments generated via Σ, return m, if and only if the majority of the Σ-commitments are commitments of $(m\|\cdot)$.

[a] We recall that the protocol requires the witness to be secret shared.

Fig. 3: $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$

6.3 The 4-Round Non-Malleable Commitment Scheme of [24]

The 4-round non-malleable commitment of Goyal et al. [24] 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 4, we recall the protocol Π_{wnmc}. This uses as an underlying building block a non-interactive commitment that is statistically bind-

PUBLIC PARAMETERS: 1^λ, ℓ_{4nmc}, tags $\mathtt{tg}_1, \ldots, \mathtt{tg}_{\ell_{\mathsf{4nmc}}}$ and a large prime q s.t. $q > 2^{\mathtt{tg}_i}$ 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 $\vec{\alpha} = (\alpha_1, \ldots, \alpha_{\ell_{\mathsf{4nmc}}})$, where $\alpha_i \in [2^{\mathtt{tg}_i}] \subset \mathbb{F}_q$.
2. Send $\vec{\alpha}, \pi_\sigma^2$ 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 $i \in [\ell_{\mathsf{4nmc}}]$, compute $a_i \leftarrow r_i\alpha_i + m$, set $\vec{a} = (a_1, \ldots, a_{\ell_{\mathsf{4nmc}}})$.
3. Send $(\pi_3^\sigma, \vec{a})$ to $\mathcal{R}_{\mathsf{wnmc}}$.

Fig. 4: $\Pi_{\mathsf{wnmc}} = (\mathcal{S}_{\mathsf{wnmc}}, \mathcal{R}_{\mathsf{wnmc}})$

ing. 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 2. *Let Σ be the Σ-commitment described in Figure 2, then $\Pi_{\mathsf{wnmc}} = (\mathcal{S}_{\mathsf{wnmc}}, \mathcal{R}_{\mathsf{wnmc}})$ described in Figure 4 enjoys the hiding property.*

This follows from Theorem 2 and from the fact that $r_1, \ldots, r_{\ell_{\mathsf{4nmc}}}$ and $a_1, \ldots, a_{\ell_{\mathsf{4nmc}}}$ information theoretically hide the committed message.

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

An informal overview of our 4-round NM commitment is given in the Introduction. Here we provide a formal description of the protocol Π_{nmc} presented in Figure 5. We conclude this section with a sketch of the proof.

7.1 Formal Description of $\Pi_{\mathsf{nmc}} = ((\mathcal{S}_{\mathsf{nmc}}, \mathcal{R}_{\mathsf{nmc}}), \mathsf{Dec}_{\mathsf{nmc}})$

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.

- A 3-round public-coin delayed-input adaptive-input SHVZK commit-and-prove protocol $\Pi_{\mathsf{tr}} = (\mathcal{P}_{\mathsf{tr}}, \mathcal{V}_{\mathsf{tr}})$ (as defined in Figure 3) for the relation $\mathsf{Rel}_{\mathsf{tr}} = \{((m_0, m_1), w) : m_0 = w \vee m_1 = w\}$. We denote the adaptive-input SHVZK simulator with $\mathsf{Sim}_{\mathsf{tr}}$.

COMMON INPUTS: 1^λ, parameters $k, n, t, \ell_{\mathsf{nmc}}$, tags $\mathtt{tg}_1, \ldots, \mathtt{tg}_\ell$ and a large prime q s.t. $q > 2^{\mathtt{tg}_i}$ for all i.
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 runs $\mathcal{P}_{\mathsf{comWI}}$ on input $(1^\lambda, \hat{s}_0, 0)$ 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 $r_1, \ldots, r_{\ell_{\mathsf{nmc}}}$ and run $\mathcal{P}_{\mathsf{CP}}$ on input $(1^\lambda, m||r_1||, \ldots, ||r_{\ell_{\mathsf{nmc}}})$ thus obtaining $(\pi_1, \pi_1^{\mathtt{com}})$.
   1.1. Run the simulator for Π_{tr}: Pick a random β_0 from the space of all the possible challengers of Π_{tr}, set $\pi_2^{\mathsf{tr}} = \pi_1||\beta_0$.
   1.2. Run $\mathsf{Sim}_{\mathsf{tr}}$ on input π_2^{tr} thus obtaining $(\mathsf{aux}, \pi_1^{\mathsf{tr}})$.
   1.3. Compute the second round π_2^{comWI} of $\mathcal{V}_{\mathsf{comWI}}$.
2. Send $(\pi_1^{\mathtt{com}}, \pi_1^{\mathsf{tr}}, \pi_2^{\mathsf{comWI}})$ 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 $\vec{\alpha} = (\alpha_1, \ldots, \alpha_\ell)$, 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. Sample a random β_1 from the space of all the possible challenges of Π_{tr} and send $(\vec{\alpha}, \pi_3^{\mathsf{comWI}}, \hat{s}_0, \hat{s}_1, \beta_1)$ 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. Compute the 3rd round of Π_{wnmc}: For all $i \in [\ell_{\mathsf{nmc}}]$, compute $a_i = r_i\alpha_i + m$.
3. Define $\mathsf{state} = \{a_i, \alpha_i\}_{i\in[\ell_{\mathsf{nmc}}]}$, and $w = \big(m, \{r_i\}_{i\in[\ell_{\mathsf{nmc}}]}\big)$
4. Compute the third message of Π_{CP} by running $\mathcal{P}_{\mathsf{CP}}$ on input $(\mathsf{state}, w, \pi_2 = \beta_0 \oplus \beta_1)$, obtaining $(\pi_3, \pi_3^{\mathtt{com}})$.
5. Run the simulator for Π_{tr}: Run $\mathsf{Sim}_{\mathsf{tr}}$ on input $(\mathsf{aux}, \hat{s}_0, \hat{s}_1)$ thus obtaining π_3^{tr}
6. Send $\pi_2^{\mathsf{tr}} = (\pi_1||\beta_0), \pi_3^{\mathsf{tr}}, \pi_3, (a_i)_{i\in[\ell_{\mathsf{nmc}}]}$ to $\mathcal{R}_{\mathsf{nmc}}$.

Verification step. $\mathcal{R}_{\mathsf{nmc}}$ parses π_{tr}^2 as $(\pi_1||\beta_0)$, set $\pi_2 = \beta_0 \oplus \beta_1$ and accepts the commitment if and only if[a]:

1. $\mathcal{V}_{\mathsf{tr}}, \mathcal{V}_{\mathsf{CP}}$ accept, respectively, the proofs for $(\pi_1^{\mathsf{tr}}, \pi_2^{\mathsf{tr}}, \pi_3^{\mathsf{tr}})$ and $\big((\pi_1, \pi_1^{\mathtt{com}}), \pi_2, (\pi_3, \pi^{\mathtt{com}})\big)$

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

1. $\mathcal{S}_{\mathsf{nmc}}$ sends the decommitment information for Π_{CP} for $(m||r_1||, \ldots, ||r_{\ell_{\mathsf{nmc}}})$.
2. $\mathcal{R}_{\mathsf{nmc}}$ checks if the decommitment information for Π_{CP} are valid w.r.t. the message $(m||r_1||, \ldots, ||r_{\ell_{\mathsf{nmc}}})$, and accepts m as the decommitted message if $(m||r_1||, \ldots, ||r_\ell)$ is consistent with $\{a_i, \alpha_i\}_{i\in[\ell_{\mathsf{nmc}}]}$.

[a] We abuse of notation when using the $\oplus$ symbol because the challenge π_2 it is not a binary string, but a set of indices. In our case β_0 and β_1 are just a set of indexes, and π_2 is obtained by pairwise summing (with modular arithmetic) the indices in β_0 and β_1

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

- A 3-round public-coin SHVZK, delayed-input complete commit-and-prove protocol $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ (as defined in Figure 3, but using λ^3 parallel repetitions) for the relation $\mathsf{Rel}_{\mathsf{CP}}$ defined as follows:

$$\mathsf{Rel}_{\mathsf{CP}} = \left\{ \begin{matrix} \mathtt{st} = \left(\{a_i, \alpha_i\}_{i\in[\ell_{\mathsf{nmc}}]}\right) \\ w = \left(m, \{r_i\}_{i\in[\ell_{\mathsf{nmc}}]}\right) \end{matrix} \middle| \; \forall\; i \;\in [\ell_{\mathsf{nmc}}]\; a_i = m + r_i\alpha_i \right\}.$$

- A one-of-two binding commitment scheme $\Pi_{\mathsf{comWI}} = (\mathcal{P}_{\mathsf{comWI}}, \mathcal{V}_{\mathsf{comWI}})$ (Definition 4).

The reason why we explicitly require Π_{tr} and Π_{CP} to be protocols constructed following the approach described in Section 6.2 is that in the security proof we will exploit the fact that Π_{tr} and Π_{CP} are based on non-malleable HVZK with respect to commitment protocols. We refer the reader to the full version for a thorough discussion on this and for the full proof.

Theorem 4. *Let $\Pi_{\mathsf{tr}} = (\mathcal{P}_{\mathsf{tr}}, \mathcal{V}_{\mathsf{tr}})$ be the 3-round public-coin adaptive-input SHVZK commit-and-prove for the relation $\mathsf{Rel}_{\mathsf{tr}}$, defined in Figure 3, let $\Pi_{\mathsf{CP}} = (\mathcal{P}_{\mathsf{CP}}, \mathcal{V}_{\mathsf{CP}})$ be the 3-round public-coin SHVZK commit-and-prove for the relation $\mathsf{Rel}_{\mathsf{CP}}$, defined in Figure 3, let $\Pi_{\mathsf{comWI}} = (\mathcal{P}_{\mathsf{comWI}}, \mathcal{V}_{\mathsf{comWI}})$ be the one-of-two binding commitment scheme, then $\Pi_{\mathsf{nmc}} = ((\mathcal{S}_{\mathsf{nmc}}, \mathcal{R}_{\mathsf{nmc}}), \mathsf{Dec}_{\mathsf{nmc}})$, described in Figure 5 is a 4-round non-malleable commitment.*

The corollary given below immediately follows from the results shown in the previous sections and from the fact that Π_{comWI} can be instantiated in a black-box way from one-to-one one-way functions.

Corollary 3. *Assuming the existence of one-to-one one-way functions, there exists a 4-round non-malleable commitment that makes black-box use of the OWFs.*

8 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 [25, Fig. 2] (that we also recall in Figure 4), 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 they receive 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 [25, Fig. 2] that we mention above.

Although our approach is inspired by [25], where the authors also lift the security of a weak-non-malleable commitment scheme relying on zero-knowledge, concretely, our techniques significantly depart from those of [25]. 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 [25, Fig. 2], instead of a modified version, as the authors of [25] 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, and to cope with the rewinds performed by the extractor in the proof (while still relying on the WI property of Π), the prover prepares n first rounds for the non-rewind secure WI protocol (denoted with 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. The reason is that there is a simple adversarial strategy for which such a proof 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 the authors of [25] 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, Goyal et al. rely on a more sophisticated version of the weak-non-malleable commitment described 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 [9] relies on the simpler sub-scheme of [25, Fig. 2] to obtain a 4-round concurrent non-malleable commitment scheme. To summarize, the main difference between ours and the approach of [25] (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.

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

1. Badrinarayanan, S., Goyal, V., Jain, A., Kalai, Y.T., Khurana, D., Sahai, A.: Promise zero knowledge and its applications to round optimal MPC. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 459–487. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96881-0_16
2. Barak, B.: Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In FOCS (2002)
3. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: 22nd Annual ACM Symposium on Theory of Computing. pp. 503–513. ACM Press, Baltimore, MD, USA (May 14–16, 1990). https://doi.org/10.1145/100216.100287
4. Cao, Z., Visconti, I., Zhang, Z.: Constant-round concurrent non-malleable statistically binding commitments and decommitments. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6056, pp. 193–208. Springer (2010)
5. Chatterjee, R., Liang, X., Pandey, O.: Improved black-box constructions of composable secure computation. In: Czumaj, A., Dawar, A., Merelli, E. (eds.) ICALP 2020: 47th International Colloquium on Automata, Languages and Programming. LIPIcs, vol. 168, pp. 28:1–28:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Saarbrücken, Germany (Jul 8–11, 2020). https://doi.org/10.4230/LIPIcs.ICALP.2020.28
6. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: 26th Annual Symposium on Foundations of Computer Science. pp. 383–395. IEEE Computer Society Press, Portland, Oregon (Oct 21–23, 1985). https://doi.org/10.1109/SFCS.1985.64

7. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Round optimal secure multiparty computation from minimal assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020: 18th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 12551, pp. 291–319. Springer, Heidelberg, Germany, Durham, NC, USA (Nov 16–19, 2020). https://doi.org/10.1007/978-3-030-64378-2_11
8. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part III. Lecture Notes in Computer Science, vol. 9816, pp. 270–299. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53015-3_10
9. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017, Part II. Lecture Notes in Computer Science, vol. 10402, pp. 127–157. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63715-0_5
10. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Round-optimal secure two-party computation from trapdoor permutations. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 678–710. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017). https://doi.org/10.1007/978-3-319-70500-2_23
11. Ciampi, M., Parisella, R., Venturi, D.: On adaptive security of delayed-input sigma protocols and fiat-shamir NIZKs. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20: 12th International Conference on Security in Communication Networks. Lecture Notes in Computer Science, vol. 12238, pp. 670–690. Springer, Heidelberg, Germany, Amalfi, Italy (Sep 14–16, 2020). https://doi.org/10.1007/978-3-030-57990-6_33
12. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved OR-composition of sigma-protocols. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A: 13th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 9563, pp. 112–141. Springer, Heidelberg, Germany, Tel Aviv, Israel (Jan 10–13, 2016). https://doi.org/10.1007/978-3-662-49099-0_5
13. Ciampi, M., Ravi, D., Siniscalchi, L., Waldner, H.: Round-optimal multi-party computation with identifiable abort. In: Dunkelman, O., Dziembowski, S. (eds.) 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. Lecture Notes in Computer Science, vol. 13275, pp. 335–364. Springer (2022). https://doi.org/10.1007/978-3-031-06944-4_12, https://doi.org/10.1007/978-3-031-06944-4_12
14. Dachman-Soled, D., Malkin, T., Raykova, M., Venkitasubramaniam, M.: 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. pp. 316–336 (2013)
15. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd Annual ACM Symposium on Theory of Computing. pp. 542–552. ACM Press, New Orleans, LA, USA (May 6–8, 1991). https://doi.org/10.1145/103418.103474

16. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: 31st Annual Symposium on Foundations of Computer Science. pp. 308–317. IEEE Computer Society Press, St. Louis, MO, USA (Oct 22–24, 1990). https://doi.org/10.1109/FSCS.1990.89549
17. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: 33rd Annual ACM Symposium on Theory of Computing. pp. 580–589. ACM Press, Crete, Greece (Jul 6–8, 2001). https://doi.org/10.1145/380752.380853
18. Goldreich, O.: Foundations of Cryptography: Volume 1. Cambridge University Press, New York, NY, USA (2006)
19. Goldreich, O., Levin, L.A.: 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. pp. 25–32 (1989)
20. Goyal, V.: Constant round non-malleable protocols using one way functions. In STOC (2011)
21. Goyal, V., Khurana, D., Sahai, A.: Breaking the three round barrier for non-malleable commitments. In: 57th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2016. IEEE (2016)
22. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: 53rd Annual Symposium on Foundations of Computer Science. pp. 51–60. IEEE Computer Society Press, New Brunswick, NJ, USA (Oct 20–23, 2012). https://doi.org/10.1109/FOCS.2012.47
23. Goyal, V., Richelson, S.: Non-malleable commitments using Goldreich-Levin list decoding. In: Zuckerman, D. (ed.) 60th Annual Symposium on Foundations of Computer Science. pp. 686–699. IEEE Computer Society Press, Baltimore, MD, USA (Nov 9–12, 2019). https://doi.org/10.1109/FOCS.2019.00047
24. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: 55th Annual Symposium on Foundations of Computer Science. pp. 41–50. IEEE Computer Society Press, Philadelphia, PA, USA (Oct 18–21, 2014). https://doi.org/10.1109/FOCS.2014.13
25. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. Cryptology ePrint Archive, Paper 2014/586, 2014, https://eprint.iacr.org/2014/586
26. Halevi, S., Hazay, C., Polychroniadou, A., Venkitasubramaniam, M.: Round-optimal secure multi-party computation. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 488–520. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96881-0_17
27. Hazay, C., Venkitasubramaniam, M.: On the power of secure two-party computation. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 397–429. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53008-5_14
28. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) 39th Annual ACM Symposium on Theory of Computing. pp. 21–30. ACM Press, San Diego, CA, USA (Jun 11–13, 2007). https://doi.org/10.1145/1250790.1250794
29. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography

Conference, Part II. Lecture Notes in Computer Science, vol. 10678, pp. 139–171. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017). https://doi.org/10.1007/978-3-319-70503-3_5
30. Khurana, D., Ostrovsky, R., Srinivasan, A.: Round optimal black-box "commit-and-prove". In: Beimel, A., Dziembowski, S. (eds.) TCC 2018: 16th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 11239, pp. 286–313. Springer, Heidelberg, Germany, Panaji, India (Nov 11–14, 2018). https://doi.org/10.1007/978-3-030-03807-6_11
31. Lin, H., Pass, R.: Constant-round non-malleable commitments from any one-way function. In: Fortnow, L., Vadhan, S.P. (eds.) Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011. pp. 705–714. ACM (2011)
32. Lin, H., Pass, R.: Constant-round nonmalleable commitments from any one-way function. J. ACM **62**(1), 5:1–5:30 (2015)
33. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In TCC (2008)
34. Mahmoody, M., Pass, R.: The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 701–718. Springer (2012)
35. Ostrovsky, R., Persiano, G., Visconti, I.: Simulation-based concurrent non-malleable commitments and decommitments. In: Reingold, O. (ed.) Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5444, pp. 91–108. Springer (2009)
36. Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: TCC. pp. 334–354 (2013)
37. Pass, R., Rosen, A.: 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. pp. 404–413. IEEE Computer Society (2003)
38. Pass, R., Rosen, A.: Concurrent non-malleable commitments. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings. pp. 563–572 (2005)
39. Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In STOC (2005)
40. Pass, R., Rosen, A.: Concurrent nonmalleable commitments. SIAM J. Comput. **37**(6), 1891–1925 (2008)
41. Pass, R., Rosen, A.: New and improved constructions of nonmalleable cryptographic protocols. SIAM J. Comput. **38**(2), 702–752 (2008)
42. Pass, R., Wee, H.: Constant-round non-malleable commitments from sub-exponential one-way functions. In EUROCRYPT (2010)
43. Wee, H.: 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. pp. 531–540. IEEE Computer Society (2010)
44. Yao, A.C.C.: Space-time tradeoff for answering range queries (extended abstract). In: 14th Annual ACM Symposium on Theory of Computing. pp. 128–136. ACM Press, San Francisco, CA, USA (May 5–7, 1982). https://doi.org/10.1145/800070.802185