Black-Box (and Fast) Non-Malleable Zero Knowledge

Vincenzo Botta*[1], Michele Ciampi†[2], Emmanuela Orsini‡[3], Luisa Siniscalchi§[4], and Ivan Visconti¶[1]

[1]Sapienza University of Rome, Rome, Italy.
[2]University of Edinburgh, Edinburgh, Scotland, United Kingdom.
[3]Bocconi University, Milan, Italy.
[4]Technical University of Denmark, Copenhagen, Denmark.

March 7, 2025

Abstract

Non-malleable zero-knowledge (NMZK), originally introduced in the seminal work of Dolev, Dwork, and Naor (STOC 91), is a fundamental concept for modeling the security of proof systems against man-in-the-middle attacks.

Recently, Kim, Liang, and Pandey (CRYPTO 2022) presented the first efficient constant-round NMZK argument system based solely on symmetric-key cryptography. Their construction relies on a non-black-box use of the involved cryptographic primitives and on multiple executions of Ligero (CCS 2017) that affect both the round complexity and the computational efficiency of their protocol. Their work left open the natural important challenge of achieving NMZK using the underlying primitives only in a black-box fashion (regardless of the number of rounds and actual efficiency).

In this paper, we solve the aforementioned open problem by presenting the first NMZK argument system based on the black-box use of cryptographic primitives. Our work is optimal in the use of primitives since we only need one-way functions, and asymptotically optimal in the number of rounds since we only require a constant number of rounds. Our argument system is non-malleable with respect to the strong "simulation-extractability" flavor of non-malleability.

Furthermore, we also show that our construction can be efficiently instantiated in Minicrypt, significantly improving upon the work of Kim et al., both in terms of round complexity and computational efficiency.

1 Introduction

Non-malleable zero-knowledge argument systems [DDN91] (NMZK) can be built by relying solely on one-way functions (OWFs) with only four rounds of communication [COSV17], precisely as it is known for zero-knowledge (ZK) arguments [BJY97].

However, when considering also the black-box use of the underlying cryptographic primitives the situation is very different. Indeed, while ZK arguments can be achieved in 5 (resp., 4) rounds relying on the black-box use of one-way functions (resp. 1-1 one-way functions) [KOS18, HV18], the gap in case of non-malleability is substantial since no construction is known, regardless of the round complexity and of the

*botta@di.uniroma1.it; part of the work was done while working at Warsaw University.
†michele.ciampi@ed.ac.uk
‡emmanuela.orsini@unibocconi.it
§luisi@dtu.dk
¶ivan.visconti@uniroma1.it; work mainly done while working at the University of Salerno."

used primitive. It is particularly that, instead, non-malleable commitments based on the black-box use of one-way functions exist even in constant rounds [GLOV12] (and only in 4 rounds with black-box use of 1-1 one-way functions [COS22]). Still, according to [KLP22a], black-box constructions[1] of NMZK arguments from non-malleable commitments are (surprisingly) not known.

The prior attempt of [JP14]. In [JP14], Jain and Pandey focus on the difficult problem of constructing black-box NMZK arguments, even when relying on constant-round black-box non-malleable commitment schemes. They succeed in showing a black-box construction for argument systems that is secure w.r.t. a partial notion of non-malleability, commonly referred to as *simulation soundness*. Furthermore, in the introduction of their work, they discuss the hardness of obtaining black-box NMZK and informally suggest a generic construction that uses black-box non-malleable commitments. However, they note that all proof approaches they explored were unsuccessful, leading them to abandon the protocol, therefore leaving open the question of the existence of a black-box NMZK argument system (even under the assumption of black-box non-malleable commitments).

Open problem. The state of the art leaves the following questions open.

Can we construct NMZK argument systems by only relying on the black-box use of cryptographic primitives? Can we construct them in constant rounds? Can we rely on one-way functions only?

Motivated by the theoretical interest and practical relevance of designing non-malleable schemes, we investigate the efficiency of NMZK arguments with restrictions on the available cryptographic primitives. Specifically, we focus on the recent non-black-box NMZK argument system of [KLP22b] based on one-way functions, which introduces new techniques to achieve efficiency while living in Minicrypt. The work of [KLP22b] relies on (and actually is affected by) the need of multiple executions of Ligero [AHIV17] to prove statements about computations performed during their protocol. The overhead due to multiple runs of Ligero highly dominates the overall amount of computations of their protocol and strongly affects its round complexity. This motivates the following additional open question:

If we are living in Minicrypt, can we still construct an efficient NMZK argument system therefore reducing or even avoiding the use of expensive generic ZK arguments (like Ligero)?

1.1 Our Contribution

In this work we provide a positive answer to all the above questions. We show the first NMZK argument system that only requires black-box use of cryptographic primitives. Moreover, our construction only needs one-way functions and can be instantiated in a (small) constant number of rounds. Our protocol outperforms previous work in terms of both efficiency and assumptions. In addition, it exhibits better concrete efficiency in Minicrypt when considering results described in [KLP22b]. In fact, we establish the following result.

Theorem (informal). *Assuming OWFs, there exists a* 10*-round (resp.* 9*-round) NMZK argument system that makes black-box use of OWFs (resp., 1-1 OWFs) only. Moreover, the protocol admits an efficient instantiation in Minicrypt.*

We prove the above theorem by presenting a protocol Π_{NM} which can be viewed as a compiler from 3-round public-coin special honest-verifier zero-knowledge (SHVZK) Π_{SHVZK} to NMZK. Interestingly, Π_{NM} can be seen as a concrete, specific and optimized instantiation of the generic approach proposed and discarded in [JP14]. We manage to bypass the obstacles that stopped [JP14] with a highly non-trivial proof approach. Moreover, an additional major contribution of our work is in particular due to way we manage to obtain an efficient construction. Indeed, since known constructions of black-box non-malleable commitments are not efficient enough, we embark on an even more challenging task requiring as a subprotocol a commitment scheme that

[1] In this paper when using the term "black-box" for a protocol we mean that the protocol uses the underlying cryptographic primitives in a black-box fashion. Another meaning of the term "black-box" can be associated to the ZK simulator. Both in [KLP22a] and in our construction the simulator is black-box, but we will not insist on remarking this property, therefore avoiding to include twice the keyword "black-box".

enjoys a weaker form of non-malleability. Thanks to the lighter security requirement but higher efficiency of this building block (that however introduces various other challenges to tackle in our security proofs), we will provide an efficient instantiation based on the black-box use of OWFs. [Wee10, Goy11] already proposed a weakening of non-malleable commitments to achieve a black-box extractable commitment scheme from any OWF. These commitments are used to obtain round efficient multi-party computation protocols with black-box access to 1-1 OWFs. The commitment proposed in [Wee10] has a logarithmic number of rounds, while [Goy11] proposed a notion called non-malleability w.r.t. replacement that provides security only against synchronizing adversaries. As explained later in Section 1.2, our resulting scheme Π_{NM} is significantly simpler than the protocol given by Kim et al. in [KLP22b], which we will denote by Π_{KLP} hereafter. The protocol Π_{KLP} relies on symmetric cryptographic primitives and along the way the authors designed a new primitive, called *instance-based non-malleable commitment* (IB-NMC), which can be seen as an efficient instantiation of the non-malleable commitment scheme of [GRRV14, BGR+15]. Indeed, they construct their IB-NMC scheme by modifying the scheme from [GRRV14, BGR+15], denoted as Π_{BGRRV}. This scheme itself comprises a three-round *commit phase* followed by a *proof phase* used to demonstrate the consistency of the commit phase. Π_{KLP} instantiates Π_{BGRRV} by employing, as the proof phase, an adapted version of the OR-composition protocol introduced by [CDS94], applied to two instances of the Ligero protocol [AHIV17]. However, since Ligero is not a Σ-protocol, [KLP22b] needs to instantiate a variant of the OR-composition of argument systems, incurring an additional cost in both computations and communication. Consequently, the main drawbacks that limit the efficiency of Π_{KLP} are the large round complexity and the cost of running Ligero multiple times. A crude (i.e., without trying to parallelize subprotocols as this would require a new security analysis) calculation of the number of rounds required by Π_{KLP} indicates that 4 rounds are needed for the commitment phase, more than 20 rounds are required for the proof phase and the OR composition, and an additional 4 rounds must be exchanged to fix a trapdoor statement for the prover and verifier. We refer the reader to [KLP22b] for a full description of Π_{KLP}.

We deviate from the approach of Π_{KLP} since we manage to use a subprotocol of the non-malleable commitment schemes of [GRRV14, BGR+15] that is associated to a special and partial extractor. It is special because it has some extraction capabilities against a man-in-the-middle (MiM) without rewinding the honest sender; it is partial because the quality of the extraction is not as good as that provided by an extractor of a classical extractable commitment scheme.

The subprotocol of the non-malleable commitment scheme of Π_{BGRRV} that we will use consists only of the first 3 rounds of their commit phase and that we denote by $\Pi^{3R}_{\mathsf{BGRRV}}$. This choice enables us to circumvent the complex and expensive computations of Π_{KLP}, as these computations are due to the subsequent rounds (from the 4th round onwards) of the commit phase of Π_{BGRRV}.

The special and partial extractor, given a transcript where the MiM mauls a commitment of a message m producing a well-formed commitment of a related message $\tilde{m}$, succeeds in polynomial time and with non-negligible probability in extracting $\tilde{m}$ without rewinding the honest sender. For simplicity, in this introduction we will refer to the property of a commitment scheme of admitting such a special and partial extractor as *weak non-malleability*[2]. We stress that in the formal part of the paper we will not explicitly define weak non-malleability; instead, we will formally refer to the existence of the above special and partial extractor introduced in [GRRV14, BGR+15] and used in our work. The subprotocol $\Pi^{3R}_{\mathsf{BGRRV}}$ by itself is not an extractable commitment scheme in the classical sense[3] since the special and partial extractor can fail with non-negligible probability in extracting the committed message on well-formed transcripts that can be sampled with non-negligible probability.

Our construction will require also classical extractability from such commitment scheme. Indeed, the simulator must be sure to obtain the actual message committed by the adversary. Therefore, we add a classical extractable commitment scheme Π_{3Ext} to $\Pi^{3R}_{\mathsf{BGRRV}}$ obtaining a 5-round commitment scheme that is both extractable in the classical sense and weak non-malleable (i.e., it admits a specific and partial extractor

[2]Notice that the special and partial extractor we use does not offer any guarantees when the commitment computed by the MiM is not well-formed. Therefore it is unclear if this definition implies the standard non-malleability property.

[3]A classical extractable commitment scheme requires an expected PPT extractor that works against an adversarial sender, there is no MiM, and the extractor produces transcripts identically distributed to the real game, providing when the commitment is well formed, also the committed message except with negligible probability.

that works against a MiM adversary).

Looking ahead, for generic statements, the computations required by our protocol mainly consist of three components: 1) One run of a classical 3-round public-coin SHVZK (e.g., ZKBoo/ZKB++ introduced by [GMO16, CDG+17] or Σ-protocols like Schnorr's protocol [Sch90]); 2) One run of a 3-round extractable commitment scheme Π_{3Ext}; 3) One run of the weak-non-malleable commitment scheme $\Pi^{3R}_{\mathsf{BGRRV}}$. Notice that both Π_{3Ext} and $\Pi^{3R}_{\mathsf{BGRRV}}$ require black-box use of any one-way function only, and have been used and efficiently instantiated in [KLP22b, BGR+15].

The efficiency of our protocol depends on the particular instantiation of the 3-round public-coin honest-verifier zero-knowledge Π_{SHVZK}. More concretely, if Π_{SHVZK} is ZKBoo/ZKB++, then when proving the preimage of a SHA-256 output, the cost of our construction is dominated by a single run of ZKBoo/ZKB++ on this statement; instead, to prove the same claim on a SHA-256 output, the protocol of [KLP22b] adds several executions of Ligero [AHIV17] to prove other statements[4]. When proving for instance knowledge of a discrete logarithm using Schnorr's protocol, the cost of running our scheme is dominated by Π_{3Ext} and $\Pi^{3R}_{\mathsf{BGRRV}}$ and we can completely avoid expensive tools like ZKBoo/ZKB++/Ligero. In contrast, the construction of [KLP22b] would still require, in addition to Π_{3Ext} and Π_{BGRRV}, multiple executions of Ligero on various statements that in turn impose to run the prover of Ligero in some cases and the special honest-verifier zero-knowledge simulator of Ligero in other cases. As shown in [KLP22b], the costs for these runs of Ligero strongly dominate all other costs.

Finally, without specific optimizations, we notice that our construction requires at most 9 rounds while the one of [KLP22b] requires more than 20 rounds. We give a detailed comparison in terms of efficiency between our scheme Π_{NM} and Π_{KLP} in Section 1.3. In particular, we show that to verify one instance of a SHA-256 preimage with 40-bit of statistical security, our scheme requires less than 100ms for the prover and less than 5MB of communication. This represents a 15× improvement in computation time and a 4× improvement in communication over Π_{KLP} which requires 20MB of communication and 1680ms of running time for the same circuit and security level. Finally, it is natural to expect that our significantly better round complexity would highly speed up the execution of the protocol when run on the Internet.

1.2 Overview of Techniques

We first propose a simple but insecure protocol, explaining why it fails. This is instrumental to better understand our more elaborated construction and its security properties.

1.2.1 A naïve protocol.

We start with a 3-round public-coin special honest-verifier zero-knowledge (SHVZK) proof of knowledge Π_{SHVZK} for $x \in \mathcal{L}$, where x is the common input and (π_1, π_2, π_3) is the transcript of an execution of Π_{SHVZK}. The classical approach to make such a protocol secure against malicious verifiers consists of allowing the simulator (that does not know the witness for x) to decide the challenge π_2 of the transcript (π_1, π_2, π_3) upfront. This can be achieved by computing π_2 as the xor of two sub-challenges c_0 and c_1 obtained as follows. The former, c_0, is chosen by the verifier and committed through an extractable commitment scheme Π_{3Ext} right after π_1 is played; the latter, c_1, is chosen and sent by the prover right after the commitment phase of Π_{3Ext} is over. The opening to c_0 is played right after c_1 is sent. After receiving the opening to c_0 the prover can compute and send π_3 to the verifier. In this way, the simulator will be able to decide the challenge π_2 by running the extractor of Π_{3Ext} and computing c_1 adaptively (i.e., $c_1 = \pi_2 \oplus c_0$).

While the exchange of messages just described provides zero knowledge, all the above components are obviously malleable and therefore the protocol as it is can not be a NMZK argument. A well known approach for proving non-malleability consists of showing an even stronger property called *simulation-extractability*. According to this stronger notion, it is required to show an efficient simulator that can extract a witness for the statement proved by the adversary in a so-called *right session*, while the adversary is receiving a simulated proof acting as a verifier (in a so-called *left session*). Such approach would allow claiming that

[4]Some of Ligero's executions are computed using the special honest-verifier zero-knowledge simulator, which is faster than the honest prover procedure.

the protocol satisfies simulation-extractability, which clearly implies non-malleability[5]. The extraction of a witness for the instance proved by the adversary is in theory possible by running the proof of knowledge extractor of Π_{SHVZK}. Unfortunately, this approach fails as we explain in the next paragraph.

Failure of the extractor while simulating. For simplicity, we will use γ to denote a message played in a round of the left session (where the adversary $\mathcal{A}_{\mathsf{MiM}}$ acts as the verifier) and $\tilde{\gamma}$ to denote the message corresponding to the same round of γ that is played in the right session (where $\mathcal{A}_{\mathsf{MiM}}$ acts as the prover). We assume that Π_{SHVZK} is associated with a canonical extractor that, starting from an accepting transcript $(\tilde{\pi}_1, \tilde{\pi}_2, \tilde{\pi}_3)$ through rewinds, tries to obtain a constant number of additional pairs $(\tilde{\pi}_2^i, \tilde{\pi}_3^i)$, so that $(\tilde{\pi}_1, \tilde{\pi}_2^i, \tilde{\pi}_3^i)$ is also accepting and values $\tilde{\pi}_2^i$ are all distinct with respect to each other and to $\tilde{\pi}_2$. We need to argue that the above extractor is successful, even in the event where in the left session we are simulating the messages of Π_{SHVZK}. In particular, note that the simulator decides the challenge π_2 (used to run the SHVZK simulator of Π_{SHVZK} to obtain (π_1, π_2, π_2)) and will *force* π_2 in all the simulated transcripts as described above. In order to force the challenge π_2, the simulator does the following. Upon receiving the extractable commitment, it extracts the underlying message (let us say c_0) and sends $c_1 = c_0 \oplus \pi_2$ to the receiver (the adversary in this case).

Since all components of the above protocol are malleable, an adversary could mimic what the simulator does and, in turn, the adversary could manage to force the same value $\tilde{\pi}_2$ in all the runs (i.e., $\tilde{\pi}_2^i = \tilde{\pi}_2$) invoked by the extractor on the right session, thus preventing the completion of the extraction.

1.2.2 Our NMZK Argument Π_{NM} via Our Commitment Scheme Π_{5Ext}.

To solve the above malleability issue, we design our NMZK argument Π_{NM} replacing the extractable commitment scheme Π_{3Ext} used by the verifier to send the commitment of a share $\tilde{c}_0$ of $\tilde{\pi}_2$ with a new commitment scheme with special properties. Specifically, we start with scheme given by the first three rounds of Π_{BGRRV} [BGR+15], which we denote by $\Pi^{3R}_{\mathsf{BGRRV}}$. This is a 3-round commitment scheme that has the special and partial extractor discussed earlier and we informally call weak non-malleability the corresponding property.

Note that $\Pi^{3R}_{\mathsf{BGRRV}}$ is not an extractable commitment in the classical sense and this limitation would hurt the simulator. As such, we will enhance $\Pi^{3R}_{\mathsf{BGRRV}}$ by adding also a run of Π_{3Ext} to it, so that the resulting 5-round commitment scheme, that we denote with Π_{5Ext}, is both extractable in the classical sense and enjoys the special and partial extractability property (we refer the reader to the next paragraph for a more detailed description of Π_{5Ext}).

We can now return to the previous attempt of relying on the canonical extractor of Π_{SHVZK}. While trying to obtain an additional accepting transcript $\tilde{\pi}_1, \tilde{\pi}'_2, \tilde{\pi}'_3$ with a new $\tilde{\pi}'_2$, a new sub-challenge $\tilde{c}'_0$ will be played (recall that the extractor of Π_{NM} acts as a verifier, and as such, it can sample $\tilde{c}'_0$). In more detail, the extractor of Π_{NM} completes a full execution on the right session committing via Π_{5Ext} to a random string $\tilde{c}_0$. Upon collecting the accepting transcript $\tilde{\pi}_1, \tilde{\pi}_2, \tilde{\pi}_3$ (we recall that $\tilde{\pi}_2 = \tilde{c}_0 \oplus \tilde{c}_1$), the extractor of Π_{NM} rewinds the adversary and changes the message committed via Π_{5Ext}, from $\tilde{c}_0$ to a new random $\tilde{c}'_0$, and completes this execution collecting the new transcript $\tilde{\pi}_1, \tilde{\pi}'_2, \tilde{\pi}'_3$. We need to argue that $\tilde{\pi}'_2 \neq \tilde{\pi}_2$, as this would allow the extractor of Π_{NM} to invoke the underlying extractor of Π_{SHVZK}, enabling the extraction of the witness for the statement proven in the right session by $\mathcal{A}_{\mathsf{MiM}}$.

The event where the MiM $\mathcal{A}_{\mathsf{MiM}}$ of Π_{NM} manages with non-negligible probability to choose the sub-challenge $\tilde{c}'_1$ so that $\tilde{\pi}'_2 = \tilde{c}'_0 \oplus \tilde{c}'_1 = \tilde{\pi}_2$ (i.e., the challenge of Π_{SHVZK} does not change during the rewinds) can now be leveraged to contradict the hiding of Π_{5Ext} using its weak non-malleability (i.e., relying on the existence of a special and partial extractor that outputs with non-negligible probability the message committed by a MiM without rewinding the honest sender). Indeed, we will show that the above successful $\mathcal{A}_{\mathsf{MiM}}$ can be embedded in a successful MiM of Π_{5Ext} and this let us rely on the special and partial extractor.

The reduction to the hiding of Π_{5Ext} includes a few values such as $\pi_1, \pi_2, \tilde{\pi}_1, \tilde{\pi}_2$ that ensure that in the end, with non-negligible probability, it will happen that $\tilde{c}'_1$ is such that $\tilde{\pi}'_2 = \tilde{c}'_0 \oplus \tilde{c}'_1 = \tilde{\pi}_2$. Notice that π_1

[5]This holds since the simulator with access to the adversary is a stand-alone adversary that can prove the statement to an external verifier using the extracted witness.

and $\tilde{\pi}_1$ are known before Π_{5Ext} on the right session starts, and the pair $(\pi_2, \tilde{\pi}_2)$ exists by contradiction since otherwise, the adversary would force the same challenge in the right session with negligible probability only, in which case we would not even need this reduction.

The reduction itself runs a modified experiment that does not use the extractor of the classical extractability property of Π_{5Ext}, but it uses instead the special and partial extractor associated to the weak non-malleability of Π_{5Ext}. Moreover, the reduction stops the execution of Π_{NM} when the adversary sends $\tilde{c}_1$ since this provides enough information to break the hiding of Π_{5Ext}.

The goal of forcing a specific value for π_2 on the left will be achieved by attempting an extraction of c_0 through the special and partial extractor of Π_{5Ext}. While this partial extractor is not guaranteed to succeed with overwhelming probability, it outputs the correct c_0 with non-negligible probability, and this is sufficient for our purposes. The reduction embeds $\mathcal{A}_{\mathsf{MiM}}$ and starts its interaction with the challenger of the hiding of Π_{5Ext}, picking the two different challenge messages m_0, m_1 that are two random λ-bit strings. The challenger now samples a bit $b \leftarrow \{0,1\}$ and commits to m_b. The reduction now acts as a proxy between the messages generated by the challenger and those generated by $\mathcal{A}_{\mathsf{MiM}}$ with respect to Π_{5Ext}. Upon receiving $\tilde{c}'_1$ from the adversary, the reduction computes $\tilde{\pi}_2 \oplus \tilde{c}'_1 = \mathsf{candidate}$. We now observe the following. We assumed that, conditioned on π_2 being the challenge on the left, with non-negligible probability $\mathcal{A}_{\mathsf{MiM}}$ manages to send a value $\tilde{c}'_1$, such that the xor of $\tilde{c}'_1$ with the message committed by the challenger via Π_{5Ext} is equal to $\tilde{\pi}_2$. Notice that π_2 will be the actual challenge of Π in the left session with non-negligible probability since the partial extractor succeeds extracting c_0 with non-negligible probability. In turn, this will imply that also $\widetilde{\pi}^2$ will be the challenge of Π in the right session and thus, $\mathsf{candidate}$, with non-negligible probability, represents the message committed by the challenger of hiding (i.e., $m_b = \mathsf{candidate}$). Note that $\mathsf{candidate}$ may, (still with non-negligible but not overwhelming probability) differ from both m_0 and m_1. In this case (i.e., $\mathsf{candidate}$ is neither equal to m_0 nor m_1), the reduction will make a random guess. Note that whenever $\mathsf{candidate}$ is instead equal to $m_{b'}$ (with $b' \in \{0,1\}$), the reduction is sure that $b' = b$. This is because m_0 and m_1 are two random strings unknown to $\mathcal{A}_{\mathsf{MiM}}$, hence, the probability that the challenger is committing to m_b, and we instead obtain $\mathsf{candidate} = m_{1-b}$ is negligible (i.e., the adversary can only hope to guess m_{1-b}, as m_{1-b} does not appear in the view of the adversary, but this is going to be negligible when the message space is large enough). For these reasons, we can then conclude that our reduction is successful.

Summing up, we rely on the classical extractability property of a commitment scheme to obtain a transcript that is indistinguishable from a real game. Next, we rely on the classical witness extraction procedure consisting of changing $\tilde{\pi}_2$ on the right while forcing the same π_2 on the left, and if this succeeds, the simulator-extractor ends adding to the above transcript also a correct witness. If the simulator-extractor fails, then through hybrid games we can show that the failure of the simulator-extractor can be reduced to an adversary breaking the hiding of Π_{5Ext}. This reduction will use the weak non-malleability of Π_{5Ext}.

Finally, we observe that a crucial reason why weak-non-malleability of the commitment scheme suffices is that a successful adversary of Π_{NM} must complete the right session committing and correctly opening the commitment. Therefore a malleability attack producing a badly-formed commitment (that is a commitment for which the special and partial extractor would fail) would not correspond to a succeeding adversary in our NMZK argument system.

1.2.3 Making $\Pi^{3R}_{\mathsf{BGRRV}}$ extractable obtaining Π_{5Ext}.

In the above discussion we have assumed that Π_{5Ext} is weak-non-malleable, and enjoys *classical extractability*. By classical extractability here we mean that, given an initial transcript of Π_{5Ext}, generated from an interaction between a corrupted sender and an honest receiver, it is possible to rewind the sender to obtain a message m, such that if the commitment generated by the sender admits a valid opening, this opening will correspond to m. If instead, the commitment does not admit an opening, then we have no guarantee about the correctness of the extracted message. $\Pi^{3R}_{\mathsf{BGRRV}}$ does not satisfy this notion of extractability. To see why this is the case, we recall how, at a very high level, $\Pi^{3R}_{\mathsf{BGRRV}}$ works. Let m be the message that we want to commit to. The sender sends a non-interactive commitment of m and of a random group element r. The receiver sends a random group element α, and in the third round, the sender replies with $a = r\alpha + m$.

Consider now a corrupted sender that computes a well-formed commitment (following the steps described

above). A candidate extractor then could work by sending a new second round α', thus obtaining a'. At this point, the extractor can interpolate the points $(a, \alpha), (a', \alpha')$ thus obtaining a message m'. Note that in the rewinding thread, the adversary could have used completely different values of r and m to compute a', hence, it can happen that the extracted value is m' with $m' \neq m$. The hiding of the non-interactive commitment prevents the extractor from understanding whether it is extracting the correct value. Hence, even if the original commitment generated by the sender is valid, the extractor we have described fails. It is not clear whether there is a succeeding extraction procedure, hence, we modify $\Pi^{3R}_{\mathsf{BGRRV}}$, to make it extractable obtaining Π_{5Ext}.

Let us now see how to add classical extractability. We modify $\Pi^{3R}_{\mathsf{BGRRV}}$ obtaining a 5-round protocol that we denote by Π_{5Ext}. This new protocol is obtained via a simple modification. We replace the non-interactive commitments of $\Pi^{3R}_{\mathsf{BGRRV}}$, with a three-round extractable commitment Π_{3Ext}. It is clear that this new commitment scheme is now extractable. In particular, the extractor of Π_{5Ext} can run the extractor of Π_{3Ext} to obtain and return m. The extraction is successfully conditioned (i.e., it outputs the committed message with non-negligible probability) on the adversarial sender providing a well-formed commitment.

We need also to argue that the obtained scheme is still weak-non-malleable (i.e., it admits a special and partial extractor). We first observe that the scheme is hiding. Then we prove that it is weak-non-malleable, via a reduction to its hiding property. The reduction works as follows, an external challenger acts as a committer against (in the left session) a MiM, and we define an extractor that extracts the message committed by MiM (on the right session) either by running the partial extractor of $\Pi^{3R}_{\mathsf{BGRRV}}$ (which exists due to the weak-non-malleability of the scheme), or by running the extractor of Π_{3Ext}. The choice of which extractors to run depends on the message schedule. In particular, there are two main cases of message schedule to consider: case 1) the right-session messages of $\Pi^{3R}_{\mathsf{BGRRV}}$ align with the messages of Π_{3Ext} in the left session; case 2) any other message schedule where case (1) does not occur.

In case (1), we can not extract from $\Pi^{3R}_{\mathsf{BGRRV}}$, as the rewinds would, in turn, rewind the messages of Π_{3Ext} generated from the challenger (which would compromise the hiding of the entire scheme). Hence, in this case, the extraction is performed using the extractor of Π_{3Ext}. Note that this does not cause a problem, as in this message schedule, the second and third rounds of Π_{3Ext} in the right session could (in the worst case) be aligned with the second and third rounds of $\Pi^{3R}_{\mathsf{BGRRV}}$ in the left session. But we will argue that this does not cause issues in the reduction as we can generate locally valid third-round messages for $\Pi^{3R}_{\mathsf{BGRRV}}$ (i.e., the third round of $\Pi^{3R}_{\mathsf{BGRRV}}$ can be randomly generated).

For all schedules that fall in (2) instead, the extraction can be performed by simply running the partial extractor of $\Pi^{3R}_{\mathsf{BGRRV}}$ (that exists from the weak-non-malleability). Due to the way we have defined (2), the rewinds performed on the right session can only rewind the messages of $\Pi^{3R}_{\mathsf{BGRRV}}$, but as discussed, this is something that $\Pi^{3R}_{\mathsf{BGRRV}}$ can deal with. This concludes this high-level description of the proof, but we refer the reader to Section 5 for a formal proof.

1.3 Efficiency and Comparison with Previous Results

Here we analyze the efficiency of our main protocol Π_{NM} and compare it to state-of-the-art NMZK argument systems. We want to stress that the goal of this section is not to provide a precise evaluation of the efficiency of our construction; instead we aim to compare it with the current state of the art.

Instantiation and efficiency of the main building blocks of Π_{NM}. At a high level, our scheme consists of two main building blocks, namely the 5-round extractable commitment scheme, Π_{5Ext}, obtained by compiling the 3-round weak-non-malleable commitment scheme of [BGR+15], that we denote by $\Pi^{3R}_{\mathsf{BGRRV}}$ (see Section 3 for a complete description), and a SHVZK protocol Π_{SHVZK} that we can instantiate for example with an optimized version of ZKBoo [GMO16, CDG+17] or Schnorr's protocol.

We examine these two components separately, and follow the analysis given in [KLP22b] for the first building block. However, as mentioned before, compared to [KLP22b], we only need to instantiate $\Pi^{3R}_{\mathsf{BGRRV}}$ deduced from the first three rounds of [BGR+15] and thus we avoid the consistency proof of the committed phase.

The perfect binding commitment scheme used in the first round of Π_{5Ext} is instantiated with Naor's

Scheme	#AESλ	#SHA-256$_\lambda$	Comm	Rounds	BB
Π_{KLP} [KLP22b]	λ	$2\|C\| + \sqrt{\|C_{\mathsf{cons}}\|} \cdot \log(\|C_{\mathsf{cons}}\|) + \sqrt{\|C_{\mathsf{eq}}\|} \cdot \log(\|C_{\mathsf{eq}}\|) + 2\|C_{\mathsf{eq}}\|$	$\lambda \cdot (\lambda + k^2 + 2\sqrt{\|C\|}) + \lambda \cdot (\sqrt{\|C_{\mathsf{cons}}\|} + 3\sqrt{C_{\mathsf{eq}}} + 2\sqrt{C_{\mathsf{SHA}}})$	> 20	✗
Π_{NM} (**our work**)	λ	$\lambda \cdot m$	$\lambda \cdot (3\lambda + k^2 + \log_2 3 + 2\lambda m)$	9	✓

Table 1: Asymptotic complexity of **Michele:** quale e' la differenza tra m e C_{mult}? [KLP22b] and our NMZK scheme. $|C|$ is the size of the circuit being proved and m is its multiplicative complexity; from [KLP22b], we have $|C_{\mathsf{cons}}| = O(k \cdot (|C_{\mathsf{add}(2\lambda)}| + 2|C_{\mathsf{mul}(2\lambda)}| + 3|C_{\mathsf{AES}(2\lambda)}))|$, where k is a parameter of Π_{BGRRV} related to the use of tags, and $|C_{\mathsf{eq}}| = |C_{\mathsf{AES}_\lambda}| + \lambda$.

commitment scheme [Nao90] for messages longer that one bit, where the PRGs used to mask the message can be implemented using AES in CTR mode. Denoting by AES_λ an evaluation of AES on a λ-bit string, the extractable commitment scheme requires roughly 3λ AES_λ in the commitment phase and 2λ AES_λ in the decommitment stage. In addition, the verifier needs to evaluate λ dot-products on vectors of length 2λ over $\mathbb{Z}_q$, with $q \approx 2^\lambda$.

For the second building block, we choose an improved version of the ZKBoo protocol described by [GMO16]. This is a 3-round SHVZK argument system based on the MPC-in-the-head paradigm of [IKOS07]. ZKBoo is only based on symmetric assumptions and has been successfully used to build efficient post-quantum secure signature schemes. We recall that ZKBoo, like almost all MPC-in-the-head protocols, require a certain number ρ of parallel repetitions to achieve the desired soundness error of $2^{-\sigma}$, where σ is the statistical security parameters.

In our analysis, we also use the version of the protocol proposed by Chase et al. [CDG+17] that presents several optimizations especially in communication complexity. As done in [KLP22b] with Ligero, we use ZKBoo figures to estimate the concrete efficiency of our protocol. Notice, both Ligero and ZKBoo instantiate their main components similarly, *i.e.*, random tapes are generated using AES in CTR mode; commitments are implemented using SHA-256, under the assumption that SHA-256 is a collision-resistant hash function and SHA-256$(r||\cdot)$ is a PRF (with key r)). Alternatively, to avoid random-oracle-based commitments we should use the commitment scheme proposed by Halevi and Micali [HM96]. In more details, we denote by $|C|$ the size of the circuit C being evaluated and by m the number of its multiplication gates; we measure the complexity of our protocol for a given circuit C in terms of number of AES_λ and SHA-256$_\lambda$, where, as previously mentioned, AES_λ (or SHA-256$_\lambda$) indicates an evaluation of the AES block cipher (resp. SHA-256) evaluations on a λ-bit string, similarly to what is done in [KLP22b]. Since the number of views needed for the MPC emulations in the ZKBoo protocol is only 3, overall this protocol requires roughly $3 \cdot m \cdot \rho$ AES_λ and $m \cdot \rho$ SHA-256$_\lambda$.

Communication Complexity. A main advantage of our approach is the very low round complexity compared to Π_{KLP}, which makes our protocol significantly better than previous ones especially in the WAN setting. Let C be a circuit over $\mathbb{Z}_{2^\ell}$, where $\ell = 2\lambda$, with $|\mathsf{inp}|$ input wires and m multiplication gates. To estimate the communication complexity, again we consider the main components of our scheme. We report the optimized overall cost of ZKBoo as given in [CDG+17]:

$$\mathsf{Cost}_{\mathsf{ZKB}} \approx \lceil \sigma(\log_2 3 - 1) \rceil \cdot \big(256 + 2\lambda + \log_2 3 + \ell(2/3|\mathsf{inp}| + m)\big)$$

In the extractable commitment step, we need to communicate approximatively $4\lambda^2$ bits for commitments and again λ^2 challenge bits.

We summarized the analysis of the costs and comparison to [KLP22b] in Table 1. While the table only considers asymptotic complexity, we can have a more concrete idea of the different complexity of the two protocols by considering the case $C=$ SHA-256 (i.e., by estimating the circuits in the table when proving

knowledge of preimages of SHA-256). We use the "Bristol fashion" circuit[6] representation for SHA-256, AES-128 and AES-256 and set $k = 32$ [KLP22b]. This will give $|C| \approx 120000$, $|C_{\mathsf{cons}}| \approx 5 \cdot 10^6$, $|C_{\mathsf{eq}}| \approx 30000$ and $m \approx 22600$. We can see that our scheme achieves a significant improvement compared to the scheme of Kim et al. with less than half rounds.

In addition to this analysis, and in support of it, we can also try to estimate the running time of our scheme by considering the running time of its main building blocks. Once again, we follow the same analysis done in [KLP22b], which also gives estimated figures by using for all the Ligero executions the figures given in [AHIV17]. Specifically, we adopt the same efficiency approximation for AES and SHA-256 as described in [KLP22b]. Additionally, we consider the single-threaded version of the implementation of ZKBoo, provided by [GMO16]. The work of Giacomelli et al. reports that proving SHA-256 takes roughly 30.81ms for the prover and 34.16ms for the verifier with communication of $\approx$ 193KB when $\sigma = 40$, and 54.63ms for the prover and 67.74ms for the verifier and requires communication of roughly 383KB when the statistical security parameter $\sigma = 80$. Consequently, using $\Pi_{\mathsf{NM}}^{\mathsf{ZKBoo}}$ for proving SHA-256 with a statistical security parameter σ set to 40 (or 80) results in a runtime of less than 100ms (or 300ms), along with communication requirements of less than 5MB. This is in stark contrast to the Π_{KLP} protocol, which necessitates approximately 1680ms (or 5000ms) and communication of around 20MB for the same security levels.

2 Preliminaries

Throughout this paper, we will use λ to denote the security parameter and $\mathsf{negl}(\lambda)$ to denote any function which tends to zero faster that λ^{-c}, for any constant c. We write $[n]$ to denote the set $\{1, \ldots, n\}$. We use the abbreviation PPT to denote probabilistic polynomial time. We use boldface to denote vectors, and $\langle \cdot, \cdot \rangle$ to denote inner product of vectors.

Let A and B be two interactive machines, we denote by (A, B) an interactive protocol between them. We denote by $\langle A(a), B(b) \rangle (x)$ the interaction between A and B on common input x and private input a for A and private input b for B. We denote by τ the transcript generated by $\langle A(a), B(b) \rangle (x)$. We denote by $Out_A(\langle A(a), B(b) \rangle (x))$ the output of A after the execution of the protocol and $Out_B(\langle A(a), B(b) \rangle (x))$ the output of B after the execution of the protocol.
We denote by $A^B(x)$ the output of A on input x and given oracle access to B.

2.1 Commitment Scheme

A commitment scheme $\Pi_{\mathtt{com}} = (\mathcal{C}, \mathcal{R})$ is a two-phase protocol between two PPT interactive algorithms, a committer $\mathcal{C}$ and a receiver $\mathcal{R}$. In the first phase, called *commit phase*, $\mathcal{C}$ on input a message m and a randomness r_c interacts with $\mathcal{R}$ on input r_r. Let $\tau = \langle \mathcal{C}(m, r_c), \mathcal{R}(r_r) \rangle$ denote the commitment transcript with committer input m, committer randomness r_c and receiver randomness r_r. In the second phase, called *decommitment phase*, the committer $\mathcal{C}$ reveals m' and $\mathcal{R}$ accepts the value committed in τ to be m' if and only if $\mathcal{C}$ proves that τ can be produced on input m'. We only consider commitment schemes where the decommitment phase consists of a single message from the committer to the receiver. Let $\mathsf{Dec}(\tau, m, r_c)$ denote the polynomial time deterministic algorithm that on input a commitment transcript τ, committer message m and randomness r_c outputs 1 or 0 to denote whether or not the decommitment was accepted. We report the classic definitions of completeness, binding and hiding. We refer the reader to [Gol01] for more details.

Definition 2.1 (Completeness)**.** *A commitment scheme $(\mathcal{C}, \mathcal{R})$ is said to be complete if for any message m, committer randomness r_c and receiver randomness r_r,* Dec *on input (τ, m, r_c), where $\tau = \langle \mathcal{C}(m, r_c), \mathcal{R}(r_r) \rangle$, outputs 1.*

Definition 2.2 (Statistical Binding)**.** *A commitment scheme $(\mathcal{C}, \mathcal{R})$ is said to be* statistically binding *if for every $\mathcal{C}^*$ there exists a negligible function ν such that $\mathcal{C}^*$ succeeds in the following game with probability at most $\nu(\lambda)$:*

[6] `https://nigelsmart.github.io/MPC-Circuits/`

- *On input the security parameter λ, $\mathcal{C}^*$ interacts with $\mathcal{R}$ in the commit stage and $\mathcal{R}$ obtains the commitment τ.*
- *$\mathcal{C}^*$ outputs pairs (m_0, r_0) and (m_1, r_1).*
- *$\mathcal{C}^*$ succeeds if $\mathsf{Dec}(\tau, m_0, r_0) = \mathsf{Dec}(\tau, m_1, r_1) = 1$ and $m_0 \neq m_1$.*

If $\nu(\lambda) = 0$ we refer to a perfectly binding commitment scheme.

Definition 2.3 (Computational Hiding)**.** *A commitment scheme $(\mathcal{C}, \mathcal{R})$ is said to be* computationally hiding *if for every* PPT *$\mathcal{R}^*$, and every two messages m_0, m_1, the view of $\mathcal{R}^*$ after a commitment phase where $\mathcal{C}$ committed to m_0 is computationally indistinguishable from the view of $\mathcal{R}^*$ after participating a commitment phase where $\mathcal{C}$ committed to m_1.*

2.2 Extractable Commitments

Informally, a commitment scheme is said to be extractable (with over-extraction) if there exists a PPT extractor that extracts the committed value conditioned on the commitment being well-formed. Formally, we report the definition of [PW09].

Definition 2.4. *Consider any statistically binding, computationally hiding commitment scheme $\Pi_{\mathsf{comExt}} = (\mathcal{C}, \mathcal{R})$. Then Π_{comExt} is said to be* extractable *if there exists an expected* PPT *oracle algorithm* Ext *(the extractor) that, given oracle access to any* PPT *committer $\mathcal{C}^*$, outputs a transcript τ and a message m such that the following holds: (i) τ is identically distributed to the view of $\mathcal{C}^*$ when interacting with an honest receiver $\mathcal{R}$ in commitment phase; (ii) the probability that τ is a well-formed transcript and $m = \perp$ is negligible; (iii) if $m \neq \perp$ then $\Pr[(\exists \tilde{m} \neq m, \tilde{r}_c) : \mathsf{Dec}(\tau, \tilde{m}, \tilde{r}_c) = 1] \leq \mathsf{negl}(\lambda)$.*

We also add the following definition that we use later in Section 5.

Definition 2.5 (2-Extractable Commitments)**.** *A 3-round (resp. 4-round, resp. 5-round) commitment scheme is said to be* 2-extractable *if, there exists a polynomial-time extractor algorithm* Ext *that given a set of 2 well-formed transcripts $\{a, c_i, z_i\}_{i\in[2]}$ (resp. $\{\gamma, a, c_i, z_i\}_{i\in[2]}$, resp. $\{\alpha, \gamma, a, c_i, z_i\}_{i\in[2]}$) of the commitment phase w.r.t. the same committed message, where for each $j, j' \in [2]$, $j \neq j'$, $c_j \neq c_{j'}$, outputs the value committed in $\{a, c_1, z_1\}$ (resp. $\{\gamma, a, c_1, z_1\}$, resp. $\{\alpha, \gamma, a, c_1, z_1\}$) except with negligible probability.*

2.3 Commitment Schemes and Man-in-the-Middle Attacks

Here we report the definition of non-malleable commitment scheme from [GRRV14]. Even though our construction will not include a non-malleable commitment, still we need to use a commitment scheme with special properties against a man-in-the-middle (MiM) adversary MIM. Indeed we need a commitment scheme for which [GRRV14, Theorem 4] holds (we report this theorem in Section 3). Non-malleable commitments are defined considering two experiments that are required to produce indistinguishable views. We will refer to a game where a distinguisher would like to guess which among the two experiments is executed as in the indistinguishability game. The MiM execution is the following. Consider a (MiM) adversary MIM that is participating in two interactions called the left and the right interactions. In the left interaction, MIM is the receiver and interacts with an honest committer $\mathcal{C}$, whereas in the right interaction, MIM is the committer and interacts with an honest receiver $\mathcal{R}$. We denote all the entities used in the right session using the tilde symbol on the corresponding entities used on the left. So, if m is the value committed by $\mathcal{C}$, $\tilde{m}$ is the value committed by MIM on the right. We assume $\mathcal{C}$ has an identity $\mathsf{id} \in \{0,1\}^k$ of its choice, for $k = \Omega(\lambda)$. At the onset of the commitment phase, $\mathcal{C}$ receives the value m in input while MIM receives an auxiliary input aux. In the left session, the MiM adversary MIM interacts with $\mathcal{C}$ receiving a commitment to message m using identity id. In the right session, MIM interacts with $\mathcal{R}$ attempting to commit to a related value $\tilde{m}$ using an identity $\tilde{\mathsf{id}}$ of its choice. If the right commitment is invalid, or undefined, the committed value is set to $\perp$. If $\mathsf{id} = \tilde{\mathsf{id}}$, we set the committed value to $\perp$ (i.e., when the adversary uses the same identity of

the honest committer the attack is invalid). Let $\mathsf{MIM}_{(\mathcal{C},\mathcal{R})}(m, \mathsf{aux})$ be the random variable that describes $(\mathsf{view}, \tilde{m})$, consisting of the values committed by MIM and MIM's view in the experiment above.

In the simulated execution, a simulator SIM directly interacts with $\mathcal{R}$. Let $\mathsf{SIM}_{(C,R)}(1^\lambda, \mathsf{aux})$ be the random variable describing $(\mathsf{view}, \tilde{m})$, given by the values committed by SIM and its output. As before, whenever SIM commits in the right interaction a commitment for which the identity is the same as one of the left interaction, the committed value is set to $\perp$. We consider one-one non-malleable commitments, where MIM participates in one left and one right interaction.

We will denote a PPT MiM adversary MIM participating in the above indistinguishability game as a *valid MiM*. We report the following definition from [GRRV14].

Definition 2.6. *A valid* PPT *MiM adversary* MIM *is* successful *if there exists a message m and a* PPT *distinguisher D such that* $\Pr[D(\mathsf{MIM}_{(\mathcal{C},\mathcal{R})}(m, \mathsf{aux})_{\mathsf{aux}\in\{0,1\}^*}) = 1] - \Pr[D(\mathsf{SIM}_{(\mathcal{C},\mathcal{R})}(1^\lambda, \mathsf{aux})_{\mathsf{aux}\in\{0,1\}^*}) = 1] \geq \frac{1}{p(\lambda)}$ *for some polynomial p and infinitely many λ.*

Given a successful MIM as described in Definition 2.6, it holds that for every successful PPT MiM adversary $\mathcal{A}$ there exists a pair of messages m_0, m_1 and a PPT distinguisher D such that $\Pr[(D(\mathsf{MIM}_{(\mathcal{C},\mathcal{R})}(m_0, \mathsf{aux})_{\mathsf{aux}\in\{0,1\}^*}) = 1] - \Pr[D(\mathsf{MIM}_{(\mathcal{C},\mathcal{R})}(m_1, \mathsf{aux})_{\mathsf{aux}\in\{0,1\}^*}) = 1] \geq \frac{1}{p(\lambda)}$ for some polynomial p and infinitely many λ.

Definition 2.7. *Let* MIM *be a valid MiM* PPT *adversary which interacts with an honest sender in the left session with tag* id *and an honest receiver in the right session with tag of his choice* $\tilde{\mathsf{id}}$ *in the execution of an n-round protocol $\Pi = (\mathcal{C}, \mathcal{R})$. Let $\tau = (com_1, \ldots, com_n, \widetilde{com_1}, \ldots, \widetilde{com_n})$ be the transcript (i.e., the messages generated in the left and right sessions) obtained at the end of such an interaction. We say that $\tau \in$ WELLF if $(com_1, com_2, \ldots, com_n)$ and $(\widetilde{com_1}, \widetilde{com_2}, \ldots, \widetilde{com_n})$ are both well-formed (i.e., both the right and left session transcripts represent commitments that admit a valid opening).*

2.4 Arguments/Proofs

Informally an interactive argument/proof system for an $\mathcal{NP}$-language $\mathcal{L}$ with associated relation $\mathsf{Rel}_\mathcal{L}$ is a pair of PPT interactive algorithms $\Pi = (\mathcal{P}, \mathcal{V})$ for which no cheating $\mathcal{P}^*$ can convince, with non-negligible probability, an honest verifier on some instance x that does not belong to $\mathcal{L}$.

We say that $\Pi = (\mathcal{P}, \mathcal{V})$ is public coin if, at every round, $\mathcal{V}$ simply tosses a predetermined number of coins (i.e., a random challenge) and sends the outcome to the prover. We denote with $\pi^0, \pi^1, \ldots, \pi^\ell$ the messages of the transcript generated by $\langle \mathcal{P}(w), \mathcal{V} \rangle$. For readability we use π^0 when the first message is sent by $\mathcal{V}$ otherwise we start to enumerate the messages from 1. Moreover we say that the transcript τ of an execution $\langle \mathcal{P}(w), \mathcal{V} \rangle(x)$ is *accepting* if $Out_\mathcal{V}(\langle \mathcal{P}(w), \mathcal{V} \rangle (x)) = 1$.

In the following, we consider the special case in which the number of rounds of Π is 3 (resp. 4), i.e. the messages of the transcript are (π^1, π^2, π^3) (resp. $(\pi^0, \pi^1, \pi^2, \pi^3)$). We recall the following definition.

Definition 2.8 (Interactive Argument/Proof System)**.** *A pair of* PPT *interactive algorithms $\Pi = (\mathcal{P}, \mathcal{V})$ constitutes a* proof *(resp.,* argument*) system for an $\mathcal{NP}$-language $\mathcal{L}$ that is associated with the relation $\mathsf{Rel}_\mathcal{L}$, if the following conditions hold:*

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

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

Definition 2.9 (SHVZK)**.** *A 3-round (resp. 4-round) proof (resp., argument) system $\Pi = (\mathcal{P}, \mathcal{V})$ as defined above, is* special honest-verifier zero knowledge *(SHVZK) if there exists a* PPT *algorithm* Sim *that for any $x \in \mathcal{L}$, where $\mathcal{L}$ is an $\mathcal{NP}$-language with the associated relation $\mathsf{Rel}_\mathcal{L}$, and any challenge π^2 (resp. any challenges (π^0, π^2)) works as follow: $(\pi^1, \pi^3) \leftarrow \mathsf{Sim}(x, \pi^2)$ (resp. $(\pi^1, \pi^3) \leftarrow \mathsf{Sim}(x, \pi^0, \pi^2)$). Furthermore,*

the distribution of the output of Sim *is (computationally) indistinguishable from the distribution of a transcript obtained when* $\mathcal{V}$ *sends* π^2 *as challenge (resp.* (π^0, π^2) *as challenges) and* $\mathcal{P}$ *runs on common input* x *and any* w *such that* $\mathsf{Rel}_{\mathcal{L}}(x, w) = 1$.

Definition 2.10 (Proof of Knowledge with Canonical Extractability)**.** *A 3-round (resp., 4-round) proof system* $\Pi = (\mathcal{P}, \mathcal{V})$ *for an* $\mathcal{NP}$*-language* $\mathcal{L}$ *associated to a relation* $\mathsf{Rel}_{\mathcal{L}}$ *as defined above, is* a proof of knowledge with canonical extractor *if there exists an expected* PPT *extractor* Ext *such that, for any* PPT *adversarial prover* $\mathcal{P}^*$ *that interacting with* $\mathcal{V}$ *produces an accepting transcript for a statement* $x \in \mathcal{L}$ *with probability* $p \geq \frac{1}{|x|^c}$ *for some constant* c*, then:*

- Ext *runs as* $\mathcal{V}$ *with* $\mathcal{P}^*$*, terminating and giving in output the transcript; if this transcript is not accepting than* Ext *stops, otherwise let* $(\pi^1, \pi_1^2, \pi_1^3)$ *(resp.* $(\pi^0, \pi^1, \pi_1^2, \pi_1^3)$*) be the accepting transcript;*
- *then* Ext *rewinding multiple times* $\mathcal{P}^*$ *and playing each time a new random value instead of* π_1^2 *with overwhelming probability obtains a constant number* k *of accepting transcripts all with the same* π^1 *(resp.* π^0, π^1*) and different challenges;*
- *obtained* k *accepting transcripts* $(x, \pi^1, \pi_i^2, \pi_i^3)_{i\in[k]}$ *(resp.* $(x, \pi^0, \pi^1, \pi_i^2, \pi_i^3)_{i\in[k]}$*) such that for each* $j, z \in [k]$*,* $j \neq z$*,* $\pi_j^2 \neq \pi_z^2$*,* Ext *outputs a witness* w *such that* $\mathsf{Rel}_{\mathcal{L}}(x, w) = 1$.

2.5 Non-Malleable Interactive Arguments/Proofs

Let $\{(\mathcal{P}_{\mathsf{id}}, \mathcal{V}_{\mathsf{id}})\}_{\mathsf{id}\in\{0,1\}^*}$ be a family of interactive argument/proof system for an $\mathcal{NP}$ language $\mathcal{L}$ with the associated relation $\mathsf{Rel}_{\mathcal{L}}$. Let $x \in \mathcal{L}$ such that $|x| = \lambda$ be the public input of the protocol and w $\mathcal{P}$'s private input such that $\mathsf{Rel}_{\mathcal{L}}(x, w) = 1$.

Let MIM be a PPT MiM adversary that is simultaneously participating in one left section with $(\mathcal{P}_{\mathsf{id}}, \mathcal{V}_{\mathsf{id}})$ and one right session with $(\mathcal{P}_{\tilde{\mathsf{id}}}, \mathcal{V}_{\tilde{\mathsf{id}}})$. MIM receives as auxiliary input $\mathsf{aux} \in \{0,1\}^*$. In the left session MIM verifies the validity of the statement x by interacting with $\mathcal{P}$ using identity id. In the right session MIM proves the validity of the statement $\tilde{x}$ (chosen adaptively by MIM) to the honest verifier $\mathcal{V}$, using identity $\tilde{\mathsf{id}}$ of MIM's choice. Let $\mathsf{view}^{\mathsf{MIM}}(x, \mathsf{aux}, \mathsf{id})$ denote the joint view of $\mathsf{MIM}(x, \mathsf{aux})$ and the honest verifier $\mathcal{V}$ when MIM is verifying a statement x in the left execution, using identity id, and proving on the right a statement $\tilde{x}$ using identity $\tilde{\mathsf{id}}$.

Definition 2.11 (Simulation-Extractability [PR08])**.** *A family of argument/proof systems* $\{(\mathcal{P}_{\mathsf{id}}, \mathcal{V}_{\mathsf{id}})\}_{\mathsf{id}\in\{0,1\}^*}$ *for an* $\mathcal{NP}$*-language* $\mathcal{L}$ *with witness relation* $\mathsf{Rel}_{\mathcal{L}}$ *is simulation extractable with tags of length* $m = m(|x|)$ *if for any MiM adversary* MIM *that participates in one left session and one right session, there exists an expected* PPT Sim *such that:*

1. *The two ensembles* $\{\mathsf{Sim}^1(x, \mathsf{aux}, \mathsf{id})\}_{x\in\mathcal{L},\mathsf{aux}\in\{0,1\}^*,\mathsf{id}\in\{0,1\}^m}$ *and* $\{\mathsf{view}^{\mathsf{MIM}}(x, \mathsf{aux}, \mathsf{id})\}_{x\in\mathcal{L},\mathsf{aux}\in\{0,1\}^*,\mathsf{id}\in\{0,1\}^m}$ *are computationally indistinguishable, where* $\mathsf{Sim}^1(x, \mathsf{aux}, \mathsf{id})$ *denotes the first output of* $\mathsf{Sim}(x, \mathsf{aux}, \mathsf{id})$.
2. *Let* $x \in \mathcal{L}$*,* $\mathsf{aux} \in \{0,1\}^*$*,* $\mathsf{id} \in \{0,1\}^m$ *and let* $(\mathsf{view}, \tilde{w})$ *denote the output of* $\mathsf{Sim}(x, \mathsf{aux}, \mathsf{id})$*. Let* $\tilde{x}$ *be the right-session instance appearing in* view *and let* $\tilde{\mathsf{id}}$ *be the identity used in the right session appearing in* view*. If the right session is accepting and* $\mathsf{id} \neq \tilde{\mathsf{id}}$*, then* $\tilde{w}$ *is such that* $(\tilde{x}, \tilde{w}) \in \mathsf{Rel}_{\mathcal{L}}$.

Remark. *Differently from [KLP22b] that uses the definition of NMZK from [PR05] which in turn is based on the definition introduced by [DDN91], we use a stronger definition named simulation extractability given in [PR08]. Notice that, as proved in [PR08, Proposition 3.6] Definition 2.11 implies the tag based NMZK definition of [PR05].*

3 Commitment Scheme $\Pi^{3R}_{\mathsf{BGRRV}} = (\mathcal{C}^{3R}_{\mathsf{BGRRV}}, \mathcal{R}^{3R}_{\mathsf{BGRRV}})$

We recall here the non-malleable commitment scheme presented in [BGR+15]. This scheme consists of a three-round public-coin commitment, followed by a zero-knowledge proof to ensure that the committed phase is well formed. The commitment phase of the non-malleable commitment scheme presented

in [BGR+15] i.e., only the commitment phase from [KLP22b], is reported in Figure 1. We denote it by $\Pi^{3R}_{\mathsf{BGRRV}} = (\mathcal{C}^{3R}_{\mathsf{BGRRV}}, \mathcal{R}^{3R}_{\mathsf{BGRRV}})$. The commitment phase of $\Pi^{3R}_{\mathsf{BGRRV}}$ makes (black-box) use of a perfectly-binding commitment scheme $\Pi = (\mathsf{Com}, \mathsf{Dec})$.

Figure 1: Description of $\Pi^{3R}_{\mathsf{BGRRV}} = (\mathcal{C}^{3R}_{\mathsf{BGRRV}}, \mathcal{R}^{3R}_{\mathsf{BGRRV}})$

PUBLIC PARAMETERS: An identity $\mathsf{id} \in \{0,1\}^k$, for $k = \Omega(\lambda)$, a large prime q, an integer ℓ, and vector spaces $V_1, \ldots, V_n \subset \mathbb{Z}_q^\ell$ derived from id. These parameters satisfy the following relation: $\ell = 2(k+1)$ and $n = k+1$.
PRIVATE INPUT: $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$ holds a private message $\boldsymbol{m} \in \mathbb{Z}_q^{\ell-1}$, where $\boldsymbol{m} = (m_1, \ldots, m_{\ell-1})$.

Commitment phase: It consists of the following steps.

Round 1 ($\mathcal{C}^{3R}_{\mathsf{BGRRV}} \to \mathcal{R}^{3R}_{\mathsf{BGRRV}}$)**.** 1. $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$ picks at random values $r_1, \ldots, r_n \in \mathbb{Z}_q$ and $s_1, \ldots, s_{\ell-1}, s'_1, \ldots, s'_n$ in $\{0,1\}^\lambda$. This defines vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{Z}_q^\ell$, where $\boldsymbol{z}_i = (r_i, \boldsymbol{m})$.

2. Let $\mathtt{cm}_m = (\mathsf{Com}(m_1; s_1), \ldots, \mathsf{Com}(m_{\ell-1}; s_{\ell-1}))$ and $\mathtt{cm}_r = (\mathsf{Com}(r_1; s'_1), \ldots, \mathsf{Com}(r_n; s'_n))$, $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$ sends $\mathtt{cm} = (\mathtt{cm}_m, \mathtt{cm}_r)$ to $\mathcal{R}^{3R}_{\mathsf{BGRRV}}$.

Round 2 ($\mathcal{R}^{3R}_{\mathsf{BGRRV}} \to \mathcal{C}^{3R}_{\mathsf{BGRRV}}$)**.** Upon receiving $\mathtt{cm}$ from $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$, $\mathcal{R}^{3R}_{\mathsf{BGRRV}}$ picks at random challenge vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, where $\alpha_i \in V_i \subset \mathbb{Z}_q^\ell$, and sends $\boldsymbol{\alpha}$ to $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$.

Round 3 ($\mathcal{C}^{3R}_{\mathsf{BGRRV}} \to \mathcal{R}^{3R}_{\mathsf{BGRRV}}$)**.** Upon receiving $\boldsymbol{\alpha}$ from $\mathcal{R}^{3R}_{\mathsf{BGRRV}}$, $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$ compute $w_i = \langle \alpha_i, \boldsymbol{z}_i \rangle \in \mathbb{Z}_q$ and send $\boldsymbol{a} = (w_1, \ldots, w_n)$.

Decommitment phase: ($\mathcal{C}^{3R}_{\mathsf{BGRRV}} \to \mathcal{R}^{3R}_{\mathsf{BGRRV}}$). $\mathcal{C}^{3R}_{\mathsf{BGRRV}}$ sends $\boldsymbol{m}$ and the values $r_1, \ldots, r_n$ and $s_1, \ldots, s_{\ell-1}, s'_1, \ldots, s'_n$ to $\mathcal{R}^{3R}_{\mathsf{BGRRV}}$. $\mathcal{R}^{3R}_{\mathsf{BGRRV}}$ checks that $\mathtt{cm}_m = (\mathsf{Com}(m_1; s_1), \ldots, \mathsf{Com}(m_{\ell-1}; s_{\ell-1}))$ and $\mathtt{cm}_r = (\mathsf{Com}(r_1; s'_1), \ldots, \mathsf{Com}(r_n; s'_n))$

In the rest of the paper we will use the following theorem proven in [GRRV14], whose key points are also proven in [BGR+15].

Theorem 3.1 ([GRRV14])**.** *Let* MIM *be a valid* PPT *MiM adversary which interacts with an honest sender in the left session with tag* id *and an honest receiver in the right session with a tag of his choice* $\tilde{\mathsf{id}}$ *in the execution of* $\Pi^{3R}_{\mathsf{BGRRV}}$*. Let* $\tau = (\mathtt{cm}, \tilde{\mathtt{cm}}, \boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}, \boldsymbol{a}, \tilde{\boldsymbol{a}})$ *denote the transcript (i.e., the messages generated in the left and right sessions) obtained at the end of such an interaction and* $\tilde{m}$ *the message committed in the right session (i.e., the message that can be successfully opened,* $\perp$ *otherwise). Let* $2\tilde{p}$ *be the non-negligible probability with which* MIM *is successful according to Definition 2.6. Then there exists a* PPT Ext *which has oracle access to* MIM *such that:*

$$\Pr[\mathsf{Ext}^{\mathsf{MIM}}(\tau) \neq \tilde{m} \mid \tau \in \mathsf{WELLF}] \leq \tilde{p},$$

where the probability is over the randomness of Ext *and the one used to sample* $\tau \in \mathsf{WELLF}$ *(see Definition 2.7 for the formal definition of* WELLF*).*

Remark on Theorem 3.1. In [GRRV14] the authors state the theorem slightly differently, requiring τ to be the transcript for which an accepting zero-knowledge proof π has been provided in the left session, proving that $(\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a})$ is well formed, and also a zero-knowledge proof $\tilde{\pi}$ is provided on the right session (again, proving that $(\tilde{\mathtt{cm}}, \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{a}})$ is well-formed). In this paper, we will not use such zero-knowledge proofs and simply require the above special and partial extractor to work correctly over only those transcripts that are well formed, letting the extractor being undefined on other transcripts. When using such a theorem, we will restrict to those runs of the special and partial extractor where the transcript given as part of the input does indeed belong to WELLF.

Remark on the use of $\Pi^{3R}_{\mathsf{BGRRV}}$. In $\Pi^{3R}_{\mathsf{BGRRV}}$ the commitment in the first round is computed on a message that corresponds to a tuple of elements in $\mathbb{Z}_q$, i.e., $\boldsymbol{m} = (m_1, \ldots, m_{\ell-1}) \in \mathbb{Z}_q^{\ell-1}$. Looking ahead, in our scheme Π_{NM} (described in Section 6) we will only need to commit to a single element of $\mathbb{Z}_q$ (similarly to [KLP22b, Protocol 2]). Let $m \in \mathbb{Z}_q$ be the message to be committed in Π_{NM}, we can consider $\boldsymbol{m} = (m, m_2, \ldots, m_{\ell-1})$, where $m_2, \ldots, m_{\ell-1}$ are all set to 0.

4 Extractable Commitment Scheme

We use the 3-round public-coin extractable commitment scheme $\Pi_{\mathsf{3Ext}} = (\mathcal{C}_{\mathsf{3Ext}}, \mathcal{R}_{\mathsf{3Ext}})$ presented in [PW09, Section 4]. Π_{3Ext} is a 3-round public-coin extractable perfectly binding commitment scheme that achieves 2-extractability. Π_{3Ext} is described in detail in Figure 2, where we denote by Com the commitment phase of a non-interactive perfectly binding commitment scheme. Notice that Com can be obtained relying only on one-way permutation (i.e., 1-1 OWF).

Figure 2: Description of $\Pi_{\mathsf{3Ext}} = (\mathcal{C}_{\mathsf{3Ext}}, \mathcal{R}_{\mathsf{3Ext}})$

NOTATION: Let Com be a statistically-binding commitment scheme.
PUBLIC PARAMETERS: λ.
PRIVATE INPUT: $\mathcal{C}_{\mathsf{3Ext}}$ holds a private message $\boldsymbol{m} \in \{0,1\}^\lambda$.

Commitment phase: It consists of the following steps.

Round 1 ($\mathcal{C}_{\mathsf{3Ext}} \to \mathcal{R}_{\mathsf{3Ext}}$). $\mathcal{C}_{\mathsf{3Ext}}$ commits using Com to λ pairs of strings $\{(\boldsymbol{v}_i^0, \boldsymbol{v}_i^1)\}_{i\in[\lambda]}$, such that, for each $i \in [\lambda]$, $(\boldsymbol{v}_i^0, \boldsymbol{v}_i^1) = (\boldsymbol{\nu}_i, \boldsymbol{m} \oplus \boldsymbol{\nu}_i)$ and $\boldsymbol{\nu}_i$ are random strings in $\{0,1\}^{|\boldsymbol{m}|}$.
Let $\boldsymbol{c}$ be the list of the resulting λ pairs of commitments, $\mathcal{C}_{\mathsf{3Ext}}$ sends $\boldsymbol{c}$ to $\mathcal{R}_{\mathsf{3Ext}}$.

Round 2 ($\mathcal{R}_{\mathsf{3Ext}} \to \mathcal{C}_{\mathsf{3Ext}}$). Upon receiving $\boldsymbol{c}$ from $\mathcal{C}_{\mathsf{3Ext}}$, $\mathcal{R}_{\mathsf{3Ext}}$ sends a random challenge $\boldsymbol{e} = (e_1, \ldots, e_\lambda)$, with each $e_i \in \{0,1\}$, for $i \in [\lambda]$.

Round 3 ($\mathcal{C}_{\mathsf{3Ext}} \to \mathcal{R}_{\mathsf{3Ext}}$). Upon receiving $\boldsymbol{e}$ from $\mathcal{R}_{\mathsf{3Ext}}$, $\mathcal{C}_{\mathsf{3Ext}}$ opens the commitments $\boldsymbol{v}_i^{e_i}, i \in [\lambda]$.

Decommitment phase: ($\mathcal{C}_{\mathsf{3Ext}} \to \mathcal{R}_{\mathsf{3Ext}}$): $\mathcal{C}_{\mathsf{3Ext}}$ sends $\boldsymbol{m}$ and the opening for all λ pairs of strings. $\mathcal{R}_{\mathsf{3Ext}}$ checks that $\boldsymbol{m} = \boldsymbol{v}_1^0 \oplus \boldsymbol{v}_1^1 = \ldots = \boldsymbol{v}_\lambda^0 \oplus \boldsymbol{v}_\lambda^1$.

Theorem 4.1 ([PW09])**.** *Let Com be a perfectly binding non-interactive commitment scheme obtained from 1-1 OWF, then Π_{3Ext} is a 3-round public-coin extractable perfectly binding commitment scheme that achieves 2-extractability.*

Notice that it is possible to rely only on OWF substituting Com with a statistically binding 2-round commitment scheme from OWF. In this case, the resulting protocol is a 4-round public-coin extractable statistically-binding commitment scheme that achieves 2-extractability. In the following sections, every time that we refer to a perfectly-binding non-interactive commitment scheme Com that relies only on 1-1 OWF it is possible to substitute it with a 2-round statistically-binding commitment scheme assuming only OWF.

5 Our 5-Round Extractable Commitment Scheme Π_{5Ext}

In this section, we construct a 5-round extractable commitment scheme $\Pi_{\mathsf{5Ext}} = (\mathcal{C}, \mathcal{R})$ that satisfies Theorem 3.1 as $\Pi^{3R}_{\mathsf{BGRRV}}$ in Section 3. We use the following building blocks:

- The 3-round 2-extractable, commitment scheme Π_{3Ext} of Section 4.
- The 3-round public-coin, commitment scheme $\Pi^{3R}_{\mathsf{BGRRV}}$ of [BGR+15], and Ext of Theorem 3.1. We recall $\Pi^{3R}_{\mathsf{BGRRV}}$ in Section 3, as we are going to use its components in the design of our Π_{5Ext}.

- A non-interactive perfect binding commitment $\Pi_{\mathtt{com}} = (\mathsf{Com}, \mathsf{Dec})$.

At a very high-level Π_{5Ext}, shown in Figure 3, follows the commitment phase of $\Pi^{3R}_{\mathsf{BGRRV}}$ (described in Section 3), which in the first round takes as input a vector of $\ell - 1$ elements and commits to it (using Com) component-wise. Our only modification is that we commit to the first component of this vector using Π_{3Ext}.

Notice that even though we give here a full description of $\Pi_{\mathsf{5Ext}} = (\mathcal{C}, \mathcal{R})$, part of it is useful in the performance analysis given in Section 1.3, part of it is explicitly used in our security proof and part of it is implicitly used when referring to the security proofs given in [BGR+15].

Figure 3: Description of $\Pi_{\mathsf{5Ext}} = (\mathcal{C}, \mathcal{R})$

NOTATION: We denote by $(\mathsf{ext}_1, \mathsf{ext}_2, \mathsf{ext}_3)$ the 3 rounds of Π_{3Ext}. We denote by $(\mathtt{cm} = (\mathtt{cm}_m, \mathtt{cm}_r), \boldsymbol{\alpha}, \boldsymbol{a})$ the 3 messages of $\Pi^{3R}_{\mathsf{BGRRV}}$. Notice that the scheme has a tag id as public parameters, it is used to run the underlying $\Pi^{3R}_{\mathsf{BGRRV}}$, for readability, we do not report this input in the algorithms of $\Pi^{3R}_{\mathsf{BGRRV}}$.
PUBLIC PARAMETERS: Same as in $\Pi^{3R}_{\mathsf{BGRRV}}$, i.e. an identity $\mathsf{id} \in \{0,1\}^t$, for $t = \Omega(\lambda)$, q, $\ell = 2(t+1)$, $n = t + 1$ and λ.
$\mathcal{C}$'S PRIVATE INPUT: $m \in \mathbb{Z}_q$.

Commitment phase: It consists of the following steps.

Round 1 ($\mathcal{C} \to \mathcal{R}$).

On input the message m to commit, $\mathcal{C}$ sets $\boldsymbol{m} = (m, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{\ell-1})$, where $\boldsymbol{m}_i = 0$ for each $i \in \{2, \ldots, \ell - 1\}$.

$\mathcal{C}$ picks at random values $r_1, \ldots, r_n \in \mathbb{Z}_q$ and $s_2, \ldots, s_{\ell-1}, s'_1, \ldots, s'_n$ in $\{0,1\}^\lambda$. This defines vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{Z}_q^\ell$ where $\boldsymbol{z}_i = (r_i, \boldsymbol{m})$.

$\mathcal{C}$ computes the first round of Π_{3Ext} w.r.t. message m obtaining ext_1.

Let $\mathtt{cm}_m = (\mathsf{ext}_1, \mathsf{Com}(\boldsymbol{m}_2; s_2), \ldots, \mathsf{Com}(\boldsymbol{m}_{\ell-1}; s_{\ell-1}))$ and $\mathtt{cm}_r = (\mathsf{Com}(r_1; s'_1), \ldots, \mathsf{Com}(r_n; s'_n))$; $\mathcal{C}$ sends $\mathtt{cm} = (\mathtt{cm}_m, \mathtt{cm}_r)$ to $\mathcal{R}$.

Round 2 ($\mathcal{R} \to \mathcal{C}$).

Upon receiving $\mathtt{cm}$, $\mathcal{R}$ computes the 2nd round $\boldsymbol{\alpha}$ of $\Pi^{3R}_{\mathsf{BGRRV}}$, namely $\mathcal{R}$ picks at random a challenge vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, where $\alpha_i \in V_i \subset \mathbb{Z}_q^\ell$, and sends $\boldsymbol{\alpha}$ to $\mathcal{C}$.

Round 3 ($\mathcal{C} \to \mathcal{R}$).

Upon receiving $\boldsymbol{\alpha}$, $\mathcal{C}$ computes the 3rd round $\boldsymbol{a}$ of $\Pi^{3R}_{\mathsf{BGRRV}}$, namely $\mathcal{C}$ sends $\boldsymbol{a} = (w_1, \ldots, w_n)$, where for $i \in [n]$, $w_i = \langle \alpha_i, \boldsymbol{z}_i \rangle \in \mathbb{Z}_q$.

Round 4 ($\mathcal{R} \to \mathcal{C}$).

$\mathcal{R}$, upon receiving $\boldsymbol{a}$, computes the second round ext_2 of Π_{3Ext}. $\mathcal{R}$ sends ext_2 to $\mathcal{C}$.

Round 5 ($\mathcal{C} \to \mathcal{R}$).

Upon receiving ext_2, $\mathcal{C}$ computes the third round ext_3 of Π_{3Ext} obtaining decommitment information $\mathtt{dec}_{\mathsf{ext}}$. $\mathcal{C}$ sends ext_3 to $\mathcal{R}$.

Decommitment phase ($\mathcal{C} \to \mathcal{R}$): $\mathcal{C}$ sends to $\mathcal{R}$ the decommitment value $\mathtt{dec} = (\boldsymbol{m}, (r_1, \ldots, r_n), (\mathtt{dec}_{\mathsf{ext}}, s_2, \ldots, s_{\ell-1}, s'_1, \ldots, s'_n))$. If $((\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a}), \boldsymbol{m}, (r_1, \ldots, r_n), (\mathtt{dec}_{\mathsf{ext}}, s_2, \ldots, s_{\ell-1}, s'_1, \ldots, s'_n))$ is a valid decommitment for $\Pi^{3R}_{\mathsf{BGRRV}}$, then $\mathcal{R}$ obtains m.

Lemma 5.1. Π_{5Ext} *described in Figure 3 enjoys the* 2*-extractability property.*

Proof. The lemma follows immediately from the 2-extractability of Π_{3Ext}. □

Theorem 5.2. Π_{5Ext} *described in Figure 3 is a perfectly binding, computationally hiding 5-round extractable commitment scheme.*

Proof. The perfect binding property of Π_{5Ext} follows from the perfect binding property of $\Pi^{3R}_{\mathsf{BGRRV}}, \Pi_{\mathsf{3Ext}}$ and $\Pi_{\mathtt{com}}$. The hiding property of Π_{5Ext} follows from the hiding of $\Pi^{3R}_{\mathsf{BGRRV}}, \Pi_{\mathsf{3Ext}}$ and $\Pi_{\mathtt{com}}$. The extractability follows from the extractability of Π_{3Ext}. Indeed, we can run the extractor of Π_{3Ext} thus obtaining m. □

Theorem 5.3. *Let* MIM *be a valid* PPT *MiM adversary which interacts with an honest sender in the left session with tag* id *and an honest receiver in the right session with tag of his choice* $\tilde{\mathsf{id}}$ *in the execution of* Π_{5Ext} *(Figure 3). Let* τ *denote the transcript obtained at the end of such an interaction and* $\tilde{m}$ *the message committed in the right session (i.e., the message that can be successfully opened,* $\perp$ *otherwise). Let* $2\tilde{p}$ *be the probability with which* MIM *is successful in the indistinguishability game according to Definition 2.6, for some non-negligible* $\tilde{p} = \tilde{p}(\lambda)$*. Then there exists a non-negligible probability* $\tilde{q} = \tilde{q}(\lambda)$ *and a* PPT $\mathsf{Ext}_{\mathsf{5Ext}}$ *which has oracle access to* MIM *such that:*

$$\Pr[\mathsf{Ext}^{\mathsf{MIM}}_{\mathsf{5Ext}}(\tau) = \tilde{m} \mid \tau \in \mathsf{WELLF}] \geq \tilde{q},$$

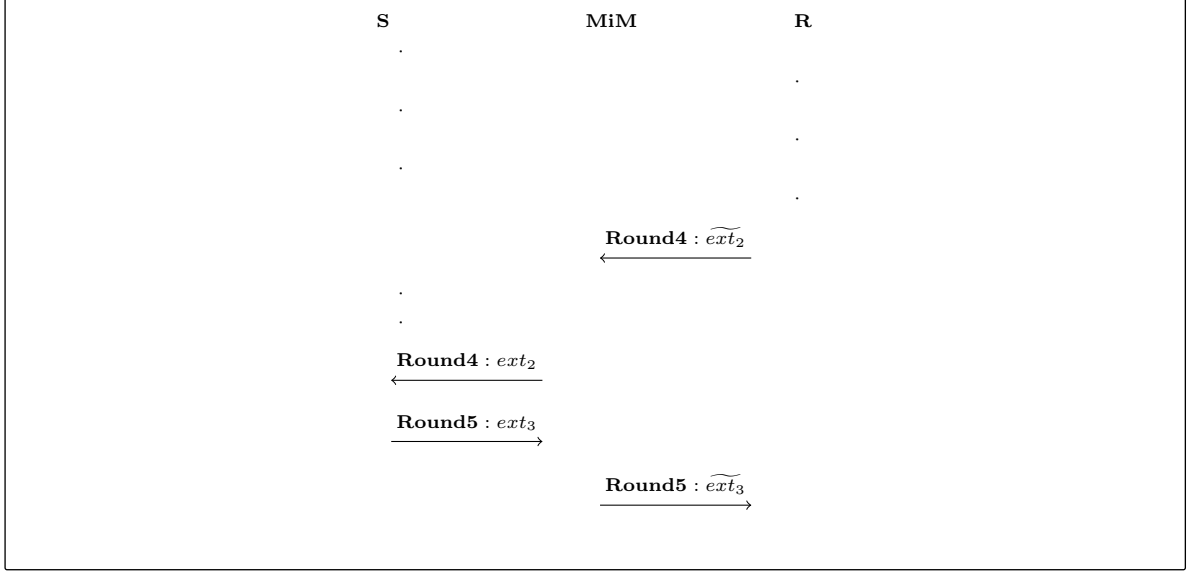
where WELLF *is the set of well-formed transcripts, the probability is over the randomness of* $\mathsf{Ext}_{\mathsf{5Ext}}$ *and the ones used to sample* $\tau \in \mathsf{WELLF}$ *which is the set of well-formed transcripts of the commitment phase.*

The high-level overview of the proof is presented at the end of Section 1.2, while the formal proof follows.

Proof. In this proof, we assume without loss of generality that the man-in-the-middle adversary MIM is non-aborting with probability at least $\widetilde{p}$. Let $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ be the extractor defined in Theorem 3.1 associated to $\Pi^{3R}_{\mathsf{BGRRV}}$. $\mathsf{Ext}_{\mathsf{5Ext}}$ on input the transcript $\tau = (\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a}, \mathsf{ext}_2, \mathsf{ext}_3, \widetilde{\mathtt{cm}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{a}}, \widetilde{\mathsf{ext}_2}, \widetilde{\mathsf{ext}_3})$ and with oracle access to the man-in-the-middle adversary MIM applies a different extracting strategy based on the schedule of the messages that yielded the generation of τ. In particular the extractor $\mathsf{Ext}_{\mathsf{5Ext}}$ distinguishes two classes of schedules:

Schedule Class of Type (a). The 4th and 5th rounds (namely, $\mathsf{ext}_2, \mathsf{ext}_3$) of the left session are both played after the 4th round (namely, $\widetilde{\mathsf{ext}_2}$) and before the 5th round (namely, $\widetilde{\mathsf{ext}_3}$)) of the right session. For clarity, we report the graphical description of this scheduling in Figure 4.

Figure 4: Schedules of type (a)



Schedule Class of Type (b). All the other types of schedules that do not belong to class (a).

We will describe how $\mathsf{Ext}_{\mathsf{5Ext}}$ extracts the committed message based on the class of schedules described above.

Type (a) In this case $\mathsf{Ext}_{\mathsf{5Ext}}$ interacts with MIM as $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ would. In a nutshell, the extractor $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ produces multiple second rounds of $\Pi^{3R}_{\mathsf{BGRRV}}$ (which are part of the 2nd round of Π_{5Ext}) in the right session and expects to receive replies to these (which corresponds to the 3rd round of Π_{5Ext}). On the left session instead, if $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ receives a new second round $\boldsymbol{\alpha}'$ it generates a new $\boldsymbol{a}'$, without the needing to know the input-randomness pair used to generate the first round $\mathtt{cm}$. To properly use the extractor $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$, we first define a valid adversary $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ for such an extractor (who receives and sends messages for the protocol $\Pi^{3R}_{\mathsf{BGRRV}}$). We define $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ via an augmented machine that internally runs MIM and filters the messages of $\Pi^{3R}_{\mathsf{BGRRV}}$. We refer to Figure 5 for the formal definition of this augmented machine/adversary, and to Figure 6 for a detailed description of how the extractor works.

We are left with arguing that $\mathsf{Ext}_{\mathsf{5Ext}}$ (Figure 6) extracts the correct committed value with non-negligible probability. If a schedule of type (a) occurs with negligible probability, then this part of the proof is trivially over. If instead, a schedule of type (a) occurs with non-negligible probability $\tilde{p}_2$, then we can observe the following. When $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ rewinds the adversary, it could happen that the schedule generated is not of type (a) anymore. This in particular means that the adversary $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ (Figure 5) may not return a reply $\tilde{\boldsymbol{a}}'$ and abort instead. Despite this, we can argue that $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ is still an adversary that does not abort with non-negligible probability. This is because we are assuming that transcript of type (a) occurs with non-negligible probability, hence our $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ is non-aborting with non-negligible probability as well. This allows us to invoke Theorem 3.1, and state that $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ is successful with non-negligible probability.

Type (b) In this case we can make use of the 2-extractability property of Π_{3Ext}. Indeed, rewinding from the 5th to the 4th rounds of Π_{5Ext}, which corresponds to the 2nd and 3rd rounds of Π_{3Ext}, we can retrieve two accepting transcripts for Π_{3Ext} in the right session which share the same first round.

To formally define this extractor, we first define in Figure 7 a valid adversarial sender $\mathcal{S}^{\mathsf{Ext}}$ for the protocol Π_{3Ext}. This new adversary internally runs MIM filtering the messages related to Π_{3Ext} in the right session. On the left session instead, $\mathcal{S}^{\mathsf{Ext}}$ sends to MIM the first round of the transcript τ

(we denoted it with $\mathtt{cm}$), generates the third round message of the left session randomly, and abort when asked to generate the last round of the left session. In particular, if MIM asks for a new 5th round message in the left session, $\mathcal{S}^{\mathsf{Ext}}$ simply does not send any message[7]. Our final extractor $\mathsf{Ext}_{\mathsf{5Ext}}$ internally runs $\mathcal{S}^{\mathsf{Ext}}$, and we formally describe it in Figure 8.

It is important to observe that in the type (b) schedule, MIM will provide a full transcript in the right session, without the need to receive the last message in the left session (that we recall, consists of the last round of Π_{3Ext}). Hence, given that type (b) transcripts appear with non-negligible probability, after the rewind performed by $\mathsf{Ext}_{\mathsf{5Ext}}$ (Figure 8), we will be again in a type (b) schedule with some non-negligible probability, hence, in this case $\mathcal{S}^{\mathsf{Ext}}$ does not abort (because it will not be asked to generate a new last round of the protocol of Π_{5Ext}).

Before concluding this part of the proof, we need to argue that MIM does not distinguish between the case when it receives well-formed messages in the third round of the left session and the case where the third round is generated randomly (i.e., inconsistently with the inputs used to generate $\mathtt{cm}$). However, if such a distinguisher exists, it would contradict the hiding of $\Pi_{\mathtt{com}}$ and Π_{3Ext}. The same argument is formalized in [GRRV14, BGR+15].

Given that MIM is providing a transcript $\tau \in \mathsf{WELLF}$ with probability $\tilde{p}$, and given that such transcript is generated via a schedule of type (b) that occurs with non-negligible probability $\tilde{p}_2 \leq \tilde{p}$, then our extractor succeeds with probability $\tilde{p}_2^2$.

Figure 5: $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$

$\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ has a left interface and a right interface where it expects to receive (and return) messages for the protocol $\Pi^{3R}_{\mathsf{BGRRV}}$. $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$ internally runs MIM (the adversary attacking Π_{5Ext}) and interacts with it in the left and right sessions as follows.

Left session

1. Upon receiving $\mathtt{cm}'$ from the left interface, send it to MIM.
2. Upon receiving $\boldsymbol{\alpha}'$ from MIM, forward the message to the left interface.
3. Upon receiving $\boldsymbol{a}'$ from the left interface, send it to MIM.
4. Upon receiving ext'_2 from MIM, if $\mathsf{ext}_2 = \mathsf{ext}'_2$ then send ext_3 to MIM, otherwise do not send any message to MIM.

Right session

1. Upon receiving $\widetilde{\mathtt{cm}'}$ from MIM, send it to the right interface.
2. Upon receiving $\tilde{\boldsymbol{\alpha}}'$ from the right interface, send it to MIM.
3. Upon receiving $\tilde{\boldsymbol{a}}'$ from MIM, send it to the right interface. Generate a random $\widetilde{\mathsf{ext}'_2}$ and send it to MIM.
4. Upon receiving $\widetilde{\mathsf{ext}_3}$ do nothing.

Figure 6: Description of $\mathsf{Ext}_{\mathsf{5Ext}}$ in case of type (a) scheduling.

INPUT: $\tau = (\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a}, \mathsf{ext}_2, \mathsf{ext}_3, \widetilde{\mathtt{cm}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{a}}, \widetilde{\mathsf{ext}_2}, \widetilde{\mathsf{ext}_3})$.

- Run $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$ with the input $(\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a}, \widetilde{\mathtt{cm}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{a}})$, giving oracle access to the adversary $\mathsf{MIM}^{3R}_{\mathsf{BGRRV}}$

[7] This is because computing a valid new last round would require knowledge of the randomness-input pair used to generate $\mathtt{cm}$, which the final extractor should not know.

(Figure 5).

- Upon receiving $\widetilde{m}$ from $\mathsf{Ext}^{3R}_{\mathsf{BGRRV}}$, return $\widetilde{m}$.

Figure 7: $\mathcal{S}^{\mathsf{Ext}}$

$\mathcal{S}^{\mathsf{Ext}}$ has a right interface where it expects to receive (and send) messages from (and to) a receiver of the protocol Π_{3Ext}. $\mathcal{S}^{\mathsf{Ext}}$ internally runs MIM (the adversary attacking Π_{5Ext}) and interacts with it in the left and right sessions as follows.

Left session

1. Send $\mathtt{cm}$ to MIM.
2. Upon receiving $\boldsymbol{\alpha}'$, if $\boldsymbol{\alpha}' = \boldsymbol{\alpha}$ then send $\boldsymbol{a}$, else sample a random $\boldsymbol{a}'$ and send it to MIM.
3. Upon receiving ext'_2 do not send any message to MIM.

Right session

1. Upon receiving $\widetilde{\mathtt{cm}'}$ from MIM, if $\widetilde{\mathtt{cm}'} = \widetilde{\mathtt{cm}}$ then send $\tilde{\boldsymbol{\alpha}}$ to MIM, else pick a random $\tilde{\boldsymbol{\alpha}}'$ and send it to MIM.
2. Upon receiving $\tilde{\boldsymbol{a}}'$ from MIM wait to receive a message $\widetilde{\mathsf{ext}'_2}$ on the right interface. Upon receiving such a message, send it to MIM.
3. Upon receiving $\widetilde{\mathsf{ext}'_3}$ from MIM, send $\widetilde{\mathsf{ext}'_3}$ to the right interface.

Figure 8: Description of $\mathsf{Ext}_{\mathsf{5Ext}}$ in case of type (b) scheduling

INPUT: $\mathsf{Ext}_{\mathsf{5Ext}}$ runs on input $\tau = (\mathtt{cm}, \boldsymbol{\alpha}, \boldsymbol{a}, \mathsf{ext}_2, \mathsf{ext}_3, \widetilde{\mathtt{cm}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{a}}, \widetilde{\mathsf{ext}_2}, \widetilde{\mathsf{ext}_3})$.

- Interact with $\mathcal{S}^{\mathsf{Ext}}$ (Figure 7) as the honest receiver for Π_{3Ext} would thus obtaining the transcript ext_1, ext_2, ext_3.
- Rewind $\mathcal{S}^{\mathsf{Ext}}$ up to the second round, and send a freshly generated ext'_2 with $\mathsf{ext}'_2 \neq \mathsf{ext}_2$.
- If $\mathcal{S}^{\mathsf{Ext}}$ does not return an accepting ext_3, then stop and return $\perp$, else obtain $\tilde{m}$ by run $\mathsf{Ext}_{\mathsf{2tran}}$ on input the transcripts $(\mathsf{ext}_1, \mathsf{ext}_2, \mathsf{ext}_3)$ and $(\mathsf{ext}_1, \mathsf{ext}'_2, \mathsf{ext}'_3)$, where $\mathsf{Ext}_{\mathsf{2tran}}$ is the extractor that exists by the 2-extractability property of Π_{3Ext}.
- Return $\tilde{m}$.

□

6 Black-Box Non-Malleable Zero Knowledge

In this section, we describe our black-box NMZK argument system Π_{NM}. Our construction provides an efficient transformation from any 3-round public-coin SHVZK proof of knowledge (with canonical extractor) to a 9-round (resp., 10-round) NMZK argument system only requiring access to 1-1 one-way functions (resp., one-way functions) in a black-box fashion. Efficient instantiations can be obtained in Minicrypt through AES and SHA-256 as already discussed in Section 1.3. The tools used in Π_{NM} are listed below:

1. The 5-round public-coin 2-extractable commitment scheme $\Pi_{\mathsf{5Ext}} = (\mathcal{C}, \mathcal{R})$ of Section 5 which satisfies Theorem 5.3 and Lemma 5.1.

2. A 3-round public-coin SHVZK proof of knowledge (with a canonical extractor) $\Pi = (\mathcal{P}, \mathcal{V})$ for the language $\mathcal{L}$ and associated relation $\mathsf{Rel}_{\mathcal{L}}$.

The protocol is described in Figure 9.

Figure 9: Description of $\Pi_{\mathsf{NM}} = (\mathcal{P}_{\mathsf{NM}}, \mathcal{V}_{\mathsf{NM}})$

NOTATION: Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a 3-round public-coin SHVZK proof of knowledge for the language $\mathcal{L}$ and associated relation $\mathsf{Rel}_{\mathcal{L}}$. Let (π^1, π^2, π^3) denote the three round messages of Π. Let Π_{5Ext} be the 5-round extractable commitment scheme defined in Section 5 which satisfies Theorem 5.3 and Lemma 5.1. Let us denote with $(com_1, com_2, com_3, com_4, com_5)$ the transcript of the commit phase of Π_{5Ext} and with dec the corresponding opening information. Let id be the identifier that will use $\mathcal{P}_{\mathsf{NM}}$. For simplicity, we will implicitly assume that the same identity id used in a session of Π_{NM} is also used for the execution of the subprotocol Π_{5Ext} inside the session of Π_{NM}.
PUBLIC PARAMETERS: $x \in \mathcal{L}$.
$\mathcal{P}_{\mathsf{NM}}$'S PRIVATE INPUT: w such that $(x, w) \in \mathsf{Rel}_{\mathcal{L}}$.

Round 1. $\mathcal{P}_{\mathsf{NM}}$ on input (x, w), computes the 1st round π^1 of Π and sends π^1 to $\mathcal{V}_{\mathsf{NM}}$.

Round 2. $\mathcal{V}_{\mathsf{NM}}$ samples $c_0 \leftarrow \{0,1\}^\lambda$ and computes the 1st round com_1 of Π_{5Ext} on input message c_0. $\mathcal{V}_{\mathsf{NM}}$ sends com_1 to $\mathcal{P}_{\mathsf{NM}}$.

Round 3. On input com_1, $\mathcal{P}_{\mathsf{NM}}$ computes the 2nd round com_2 of Π_{5Ext} and sends com_2 to $\mathcal{V}_{\mathsf{NM}}$.

Round 4. On input com_2, $\mathcal{V}_{\mathsf{NM}}$ computes the 3rd round com_3 of Π_{5Ext} and sends com_3 to $\mathcal{P}_{\mathsf{NM}}$.

Round 5. On input com_3, $\mathcal{P}_{\mathsf{NM}}$ computes the 4th round com_4 of Π_{5Ext} and sends com_4 to $\mathcal{V}_{\mathsf{NM}}$.

Round 6. On input com_4, $\mathcal{V}_{\mathsf{NM}}$ computes the 5th round com_5 and corresponding opening information dec of Π_{5Ext}w.r.t. message c_0 (i.e., the committed message is c_0). $\mathcal{V}_{\mathsf{NM}}$ sends com_5 to $\mathcal{P}_{\mathsf{NM}}$.

Round 7. $\mathcal{P}_{\mathsf{NM}}$ samples $c_1 \leftarrow \{0,1\}^\lambda$ and sends c_1 to $\mathcal{V}_{\mathsf{NM}}$.

Round 8. $\mathcal{V}_{\mathsf{NM}}$ sends (c_0, dec) to $\mathcal{P}_{\mathsf{NM}}$.

Round 9. On input (c_0, dec), $\mathcal{P}_{\mathsf{NM}}$ acts as follows. If dec is not a valid opening for $(com_1, com_2, com_3, com_4, com_5)$ w.r.t. committed message c_0 then $\mathcal{P}_{\mathsf{NM}}$ aborts. Otherwise, $\mathcal{P}_{\mathsf{NM}}$ sets $\pi^2 = c_0 \oplus c_1$ and computes the 3rd round π^3 of Π. $\mathcal{P}_{\mathsf{NM}}$ sends π^3 to $\mathcal{V}_{\mathsf{NM}}$.

Verification procedure. On input π^3, $\mathcal{V}_{\mathsf{NM}}$ computes $\pi^2 = c_0 \oplus c_1$. If (x, π^1, π^2, π^3) is an accepting transcript for Π, then $\mathcal{V}_{\mathsf{NM}}$ accepts, otherwise it aborts.

Theorem 6.1. *Given a 3-round public-coin SHVZK proof of knowledge $\Pi = (\mathcal{P}, \mathcal{V})$ with canonical extractor for an $\mathcal{NP}$ language $\mathcal{L}$, the 5-round public-coin extractable commitment scheme Π_{5Ext} of Section 5 (which satisfies Theorem 5.3 and Lemma 5.1), Π_{NM} (Figure 9) is a simulation-extractable argument system for $\mathcal{L}$ which makes only a black-box use of Π_{5Ext} and Π.*

Proof. The completeness of Π_{NM} follows by inspection.
COMPUTATIONAL SIMULATION EXTRACTABILITY.
The simulator $\mathsf{Sim}_{\mathsf{NM}}$ is described in Figure 10.

Figure 10: Description of the simulator $\mathsf{Sim}_{\mathsf{NM}}$

PUBLIC PARAMETERS: x, id.
$\mathsf{Sim}_{\mathsf{NM}}$, on input $x \in \mathcal{L}$ and $\mathsf{id} \in \{0,1\}^{\Omega(|x|)}$, simulates an execution between a prover (impersonated by $\mathsf{Sim}_{\mathsf{NM}}$) and MIM w.r.t. statement $x \in \mathcal{L}$, while in the right session $\mathsf{Sim}_{\mathsf{NM}}$ interacts with the MIM as a

verifier w.r.t. some statement $x^* \in \{0,1\}^{|x|}$ (chosen by MIM before the start of the protocol). We describe separately the left and the right executions since these two executions can be interleaved in any possible way.

Left Execution

In the left execution $\mathsf{Sim}_{\mathsf{NM}}$ runs the extractor of Π_{5Ext} to force a pre-selected challenge of Π to run the SHVZK simulator of Π.

Round 1: $\mathsf{Sim}_{\mathsf{NM}}$ picks a value $\pi_h^2 \leftarrow \{0,1\}^\lambda$ and computes $(\pi_h^1, \pi_h^3) \leftarrow \mathsf{Sim}_\Pi(x, \pi_h^2)$. $\mathsf{Sim}_{\mathsf{NM}}$ sends π_h^1 to MIM.

Round 3: Upon receiving com_1 from MIM, $\mathsf{Sim}_{\mathsf{NM}}$ computes the 2nd round com_2 of Π_{5Ext} and sends com_2 to MIM.

Round 5: Upon receiving com_3 from MIM, $\mathsf{Sim}_{\mathsf{NM}}$ computes the 4th round com_4 of Π_{5Ext} and sends com_4 to MIM.

Round 6: If MIM aborts before this stage then $\mathsf{Sim}_{\mathsf{NM}}$ also aborts giving in output the view of MIM. Otherwise, upon receiving com_5 from MIM continue to the following rewinding stage.

> Rewinding Stage. Let $\mathsf{Ext}_{\mathsf{5com}}$ be the extractor of Π_{5Ext} (Definition 2.4). $\mathsf{Sim}_{\mathsf{NM}}$ computes the following steps.
>
> $\mathsf{Sim}_{\mathsf{NM}}$ invokes $\mathsf{Ext}_{\mathsf{5com}}$ on input $(com_1, com_2, com_3, com_4, com_5)$. Then, $\mathsf{Sim}_{\mathsf{NM}}$ continues the execution acting as malicious committer $\mathcal{C}^*$ w.r.t. $\mathsf{Ext}_{\mathsf{5com}}$ while internally $\mathsf{Sim}_{\mathsf{NM}}$ is interacting with MIM (in the left and right executions). In particular, $\mathsf{Sim}_{\mathsf{NM}}$ acting as $\mathcal{C}^*$, receives (multiple times) a new 4th rounds of Π_{5Ext}, namely com_4', from $\mathsf{Ext}_{\mathsf{5com}}$. Upon receiving com_4', $\mathsf{Sim}_{\mathsf{NM}}$ rewinds MIM sending as a new 5th round of Π_{NM} the message com_4', then $\mathsf{Sim}_{\mathsf{NM}}$ forwards the MIM′s response (if any), com_5', to $\mathsf{Ext}_{\mathsf{5com}}$.
>
> In the end of the execution $\mathsf{Ext}_{\mathsf{5com}}$ outputs c_0^*.

Round 7: If $c_0^* \neq \perp$, then $\mathsf{Sim}_{\mathsf{NM}}$ computes $c_1 = \pi^2 \oplus c_0^*$, otherwise it computes c_1 as a random λ bit string; then it sends c_1 to MIM.

Round 9: Upon receiving (dec, c_0) from MIM, $\mathsf{Sim}_{\mathsf{NM}}$ checks that the received commitment of Π_{5Ext} opens to $c_0 = c_0^*$. If the check fails then $\mathsf{Sim}_{\mathsf{NM}}$ stops in this left session. Otherwise $\mathsf{Sim}_{\mathsf{NM}}$ sends π^3 to MIM.

Right Execution

In the right execution $\mathsf{Sim}_{\mathsf{NM}}$ acts as an honest verifier that using id commits to a message $\tilde{c}_0$ (that corresponds to the challenge that the canonical extractor Ext_Π of Π would play) until the 9th round in which $\mathsf{Sim}_{\mathsf{NM}}$ receives from MIM the 3rd round of Π. Let $(\widetilde{\pi}^1, \widetilde{com}_1, \widetilde{com}_2, \widetilde{com}_3, \widetilde{com}_4, \widetilde{com}_5, \tilde{c}_1, (\widetilde{\mathsf{dec}}, \tilde{c}_0), \widetilde{\pi}^3)$ be the messages exchanged in this execution. An Extracting Stage will be run only when the execution of MIM will be over and the transcript of the right session is accepting for a verifier. Otherwise the transcript will be given in output adding $\perp$ as extracted witness and $\mathsf{Sim}_{\mathsf{NM}}$ stops.

Extracting Stage: Let Ext_Π be the canonical extractor of Π and let k be the number of transcripts required to finally compute a witness. $\mathsf{Sim}_{\mathsf{NM}}$ continues as Ext_Π on the accepting transcript $(\widetilde{\pi}^1, \widetilde{\pi}^2, \widetilde{\pi}^3, x^*)$ acting also (along with MIM) as the prover of Π; $\mathsf{Sim}_{\mathsf{NM}}$ initializes the lists 2R, 3R to empty lists and performs the following steps.

Round 6 (rewound) $\mathsf{Sim}_{\mathsf{NM}}$ samples a random message $\tilde{c}'_0$, rewinds MIM to the second round and recomputes all the messages of Π_{5Ext} thus committing to $\tilde{c}'_0$.

Round 8 (rewound) $\mathsf{Sim}_{\mathsf{NM}}$ computes the following steps.

1. If the commitment phase of Π_{5Ext} fails or no message $\tilde{c}'_1$ is received from MIM (due to MIM aborting) then repeat the step described for Round 6 (rewound).
2. If instead the commitment phase of Π_{5Ext} is successful, and MIM does send a message $\tilde{c}'_1$, then do the following.
   (a) If $\widetilde{\pi}^{2'} = \tilde{c}'_0 \oplus \tilde{c}'_1$ is equal to any of the challenges contained in 2R concatenated with $\widetilde{\pi}^2$, $\mathsf{Sim}_{\mathsf{NM}}$ outputs $(\perp, \perp)$ and aborts.
   (b) Otherwise, $\mathsf{Sim}_{\mathsf{NM}}$ sends the opening $\widetilde{\mathsf{dec}}$ for the message $\tilde{c}'_0$ to MIM, sets as a random challenge for the extractor Ext_Π the message $\widetilde{\pi}^{2'} = \tilde{c}'_0 \oplus \tilde{c}'_1$ and adds $\widetilde{\pi}^{2'}$ to 2R.

Extracting w^*: $\mathsf{Sim}_{\mathsf{NM}}$ computes the following steps.

1. If no message is sent by MIM in Round 9 then $\mathsf{Sim}_{\mathsf{NM}}$ goes back to Round 6 (rewound).
2. If instead a message $\widetilde{\pi}^{3'}$ is sent by MIM then $\mathsf{Sim}_{\mathsf{NM}}$ checks if the transcript of Π is accepting and in this case adds $\widetilde{\pi}^{3'}$ to 3R, otherwise $\mathsf{Sim}_{\mathsf{NM}}$ (acting as a prover of Π) sends $\widetilde{\pi}^{3'}$ as 3rd round of Π to Ext_Π; in case Ext_Π stops then $\mathsf{Sim}_{\mathsf{NM}}$ aborts giving in output $(\perp, \perp)$, otherwise, $\mathsf{Sim}_{\mathsf{NM}}$ goes back to Round 6 (rewound).
3. If $|\mathsf{3R}| < k$ $\mathsf{Sim}_{\mathsf{NM}}$ goes to Round 6 (rewound).
4. Otherwise $\mathsf{Sim}_{\mathsf{NM}}$ performs the last computation of Ext_Π therefore computing on input $(\widetilde{\pi}^1, \widetilde{\pi}^2, \widetilde{\pi}^3, \mathsf{2R}, \mathsf{3R}, x^*)$ a corresponding valid witness w^*.

Output

Once the left and right executions are completed, $\mathsf{Sim}_{\mathsf{NM}}$ outputs (τ, w^*) where τ is the view of MIM in the main thread (messages of rewinds are discarded) of the left and right executions and w^* is the extracted witness.

Let us consider a MIM that in the right session proposes an instance x^*. We will prove that $\{\mathsf{Sim}^1_{\mathsf{NM}}(x, \mathsf{aux}, \mathsf{id})\}_{x\in\mathcal{L},\mathsf{aux}\in\{0,1\}^*,\mathsf{id}\in\{0,1\}^{\Omega(|x|)}}$ and $\{\mathsf{view}^{\mathsf{MIM}}(x, \mathsf{aux}, \mathsf{id})\}_{x\in\mathcal{L},\mathsf{aux}\in\{0,1\}^*,\mathsf{id}\in\{0,1\}^{\Omega(|x|)}}$ are computationally indistinguishable and only with negligible probability $\mathsf{Sim}_{\mathsf{NM}}$ will output a transcript with an accepting right session for x^* but without a corresponding witness w^*.

$\mathcal{H}_1$**:** This is equivalent to the real-world experiment among $\mathcal{P}_{\mathsf{NM}}$, MIM, and $\mathcal{V}_{\mathsf{NM}}$, where $\mathcal{P}_{\mathsf{NM}}$ proves to MIM a statement $x \in \mathcal{L}$ using a witness w and MIM proves to $\mathcal{V}_{\mathsf{NM}}$ that $x^* \in \mathcal{L}$, for $x^* \in \{0,1\}^\lambda$. If the proof of MIM is not accepting, the experiments terminates giving in output $(\tau, \perp)$. If instead the proof given by MIM is accepting, the experiment performs the Extracting Stage described in the right session of Figure 10 (i.e., the experiment acts as $\mathsf{Sim}_{\mathsf{NM}}$ in the Extracting Stage). If the Extracting Stage terminates before Step 4, the experiment ends giving in output $(\perp, \perp)$. Otherwise, the experiment ends giving in output (τ, w^*), where τ is the view of MIM in the main thread of the above execution and w^* is the

extracted witness for x^*.

$\mathcal{H}_2$: This is equal to $\mathcal{H}_1$ except that $\mathcal{P}_{\mathsf{NM}}$ runs th extractor of Π_{5Ext} in the left execution therefore, when MIM sends the last message of Π_{5Ext} the experiment executes the Rewinding Stage described in the left session of Figure 10. The same (a run of the extractor of Π_{5Ext} that involves an execution of the Rewinding Stage) happens in case the commitment sent by MIM through Π_{5Ext} changes during rewinds performed by the Extracting Stage. If the Rewinding Stage terminates with an output c_0^* and the adversary correctly opens the commitment played in the left session to a value that is different than c_0^*, then the experiment ends giving in output $(\perp, \perp)$, otherwise it continues as in $\mathcal{H}_1$ computing the output in the same way.

$\mathcal{H}_3$: This is equal to $\mathcal{H}_2$ except that in the left execution, the experiment fixes a candidate random value π^2 at the onset of the execution on the left session (i.e., before computing π^1). Let c_0^* be the value obtained from the Rewinding Stage of the left session. In Round 7 (after performing the Rewinding Stage) if $c_0^* \neq \perp$ the experiment sets $c_1 = \pi^2 \oplus c_0^*$. The rest of the experiment proceeds as in $\mathcal{H}_2$ computing the output in the same way.

$\mathcal{H}_4$: This is equal to $\mathcal{H}_3$ except that in Round 1 of the left execution the experiment computes $(\pi_h^1, \pi_h^3) \leftarrow \mathsf{Sim}_\Pi(x, \pi_h^2)$ where π_h^2 corresponds to π^2 of $\mathcal{H}_3$. The experiment sends π_h^1 in Round 1 and π_h^3 in Round 9. The rest of the experiment proceeds as in $\mathcal{H}_3$ computing the output in the same way.

Lemma 6.2. *$\mathcal{H}_1$ terminates with output $(\perp, \perp)$ only with negligible probability.*

Proof. Let us assume by contradiction that Lemma 6.2 does not hold. This implies that the Extracting Stage terminates by giving in the output $(\perp, \perp)$ with probability $p \geq \frac{1}{\lambda^c}$ for some constant c and infinitely many λ. Notice that the Extraction Stage is invoked only when an accepting transcript has been generated. Therefore we have that the probability of getting an accepting transcript is also at least $\frac{1}{\lambda^c}$ and thus, by the proof of knowledge property of Π it follows that the canonical extractor of Π succeeds giving in output a witness with overwhelming probability as long as each time the right session is completed in the Extracting Stage the challenge $\widetilde{\pi}^2$ is new. Therefore the only reason why the Extracting Stage still outputs $(\perp, \perp)$ with probability p (contradicting the above overwhelming probability of success) is due to the fact that the Extracting Stage fails in collecting $(\widetilde{\pi}^1, \widetilde{\pi}^2, \widetilde{\pi}^3, \mathsf{2R}, \mathsf{3R}, x^*)$ such that the elements in $\widetilde{\mathsf{2R}} = (\mathsf{2R} + \widetilde{\pi}^2)$ are all distinct[8]. As a consequence, with non-negligible probability at least two elements in $\widetilde{\mathsf{2R}}$ are identical. We can define the following event, which we call Bad. Let $(\widetilde{\pi}^1, \widetilde{\pi}^2, \widetilde{\pi}^3)$ be the transcript obtained before starting the Extracting Stage of the right session. Then, Bad is defined as the event that during the Extracting Stage (and thus in the presence of a different $\tilde{c}_0'$), MIM continues the execution of the right session and provides $\tilde{c}_1'$ such that $\tilde{c}_1' \oplus \tilde{c}_0'$ is equal to $\widetilde{\pi}^2$. Since with non-negligible probability at least two elements in $\widetilde{\mathsf{2R}}$ are identical, we have that Bad happens with non-negligible probability. We now use this fact to show a reduction $\mathcal{A}_{\mathsf{ExpHiding}}$ to the hiding of Π_{5Ext}, therefore reaching a contradiction. Let $\mathsf{C}_{\mathsf{ExpHiding}}$ be the corresponding challenger.

$\mathcal{A}_{\mathsf{ExpHiding}}$ interacts with MIM acting exactly as in hybrid $\mathcal{H}_1$. Upon receiving $\widetilde{\pi}^2$ from the right session $\mathcal{A}_{\mathsf{ExpHiding}}$ rewinds MIM at the onset of Round 2 in the right session of Π_{NM} and continues the execution as follows.

In the right session of Π_{NM}, $\mathcal{A}_{\mathsf{ExpHiding}}$ received a value $\widetilde{\pi}^1$ from MIM and then $\mathcal{A}_{\mathsf{ExpHiding}}$ chooses two random values $(\tilde{c}_0', \tilde{c}_0)$ and sends them to $\mathsf{C}_{\mathsf{ExpHiding}}$. At this point $\mathcal{A}_{\mathsf{ExpHiding}}$ will act as a proxy between $\mathsf{C}_{\mathsf{ExpHiding}}$ and MIM for the messages of Π_{5Ext} and continuing the rest of the experiment as before, until MIM sends $\tilde{c}_1$ in the right session. At this point, $\mathcal{A}_{\mathsf{ExpHiding}}$ checks if $\widetilde{\pi}^2 = \tilde{c}_0 \oplus \tilde{c}_1$, and in this case $\mathcal{A}_{\mathsf{ExpHiding}}$ sends $b = 0$ to $\mathsf{C}_{\mathsf{ExpHiding}}$. If $\widetilde{\pi}^2 = \tilde{c}_0' \oplus \tilde{c}_1$ $\mathcal{A}_{\mathsf{ExpHiding}}$ sends $b = 1$. Otherwise $\mathcal{A}_{\mathsf{ExpHiding}}$ sends a random bit.

Since we are assuming that Bad happens with non-negligible probability then, by noticing that the value among $\tilde{c}_0'$ and $\tilde{c}_0$ that is not committed by $\mathsf{C}_{\mathsf{ExpHiding}}$ is information theoretically hidden, we have that $\mathcal{A}_{\mathsf{ExpHiding}}$ returns the bit b chosen by $\mathsf{C}_{\mathsf{ExpHiding}}$ with probability $\frac{1}{2} + \widetilde{p}$ where $\widetilde{p}$ is non-negligible. This contradicts the hiding of Π_{5Ext}.

[8] Abusing the notation we use the symbol "+" to indicate the append operation of lists.

Finally notice that we did not make any restriction on the scheduling of messages of the right and left sessions. □

Lemma 6.3. *$\mathcal{H}_2$ is statistically indistinguishable from $\mathcal{H}_1$.*

Proof. $\mathcal{H}_2$ and $\mathcal{H}_1$ are statistically indistinguishable for the following reasons. First of all, the extractor of Π_{5Ext} produces a perfectly indistinguishable transcript of Π_{5Ext}. There can be an additional impact on the output of the experiment depending on the value extracted by the extractor of Π_{5Ext} and we distinguish the following cases. In the first case, the message committed in the left session is invalid and the extractor outputs a valid message (due to over-extraction). Notice that an invalid commitment does not admit a valid opening, and thus this case corresponds in both hybrids to the left session that reaches at most the (invalid) opening of the commitment, therefore producing no deviation in the two distributions. The second case, concerns the fact that the extractor of Π_{5Ext} outputs a legitimate message that could be different from the one actually opened by the adversary in the left session. While this makes a deviation among the two hybrids, by the statistical binding of Π_{5Ext} this can happen only with negligible probability. The third case refers to the extractor giving in output $\perp$ while instead the commitment admits a correct opening, therefore producing again a gap in the outputs of the two hybrids. However this failure in the extraction can happen by Definition 2.4 only with negligible probability.

Finally, when the extractor of Π_{5Ext} outputs the same message that is then opened by the adversary, the two hybrids produce identically distributed outputs. Also during the Extracting Stage the messages played in $\mathcal{H}_2$ are identically distributed to the ones of $\mathcal{H}_1$. More formally, as shown in the proof of Lemma 6.2 a failure in extracting a witness implies that the event Bad happens with non-negligible probability. The very same reduction shown in Lemma 6.2 can be repeated here, running $\mathcal{A}_{\mathsf{ExpHiding}}$ once the transcripts of the right session is completed, without running the extractor of Π_{5Ext} when the challenger $\mathsf{C}_{\mathsf{ExpHiding}}$ is involved. Therefore we can conclude that the outputs of the two hybrids are statistically indistinguishable.

We stress that the claim holds regardless of the scheduling of the left execution and right execution. Indeed, note that the Extracting Stage consists of playing messages that are identically distributed with respect to the ones of an honest verifier. Similarly, the Rewinding Stage consists of playing messages that are identically distributed with respect to the ones of an honest receiver. Moreover, each Stage fails only with negligible probability, and repeating it polynomially many times (as it could be required because of rewinds performed by the other stage) still leaves negligible the probability of a failure. Therefore, any possible interleaving of messages between left and right sessions, does not noticeably affect the success probabilities of the extractor of Π and the extractor of Π_{5Ext}. □

Lemma 6.4. *$\mathcal{H}_3$ is computationally indistinguishable from $\mathcal{H}_2$.*

Proof. The first output of $\mathcal{H}_3$ is identical to the one of $\mathcal{H}_2$ since selecting π^2 randomly in advance to then establish $c_1 = \pi^2 \oplus c_0$ is equivalent to randomly selecting c_1 and then computing π^2. We now focus on the second value in the output of the experiment. We consider again the event Bad that corresponds to a failure in computing a witness for the accepting transcript appearing in the right session. Showing that Bad happens with negligible probability would conclude the proof of this Lemma. Notice that the only difference between $\mathcal{H}_3$ and $\mathcal{H}_2$ is that in the left session of $\mathcal{H}_3$ the experiment sets a specific value, namely π^2, as a challenge for Π. If the event Bad happens in $\mathcal{H}_3$ with non-negligible probability then it must be the case that MIM manages to force a value $\widetilde{\pi^2}$ on the right session that will appear again during the Extracting Stage. We will contradict the hiding of Π_{5Ext}, using the fact that Π_{5Ext} satisfies Theorem 5.3.

Let π^2 be the challenge message of Π for the left session, which is computed as described in $\mathcal{H}_3$.

Suppose that Bad happens with non-negligible probability $\widetilde{p}$. Let $\widetilde{\pi^2}$ be the value that has non-negligible probability[9] to appear again in the right session during the Extracting Stage when π^2 is forced on the left session and $\widetilde{\pi^1}$ is fixed.

[9]It follows by a standard averaging argument that given the non-negligible probability with which the event Bad happens, we can fix some randomness of the experiment so that the probability that MIM will complete the right session when $\pi^1, \tilde{\pi}^1, \tilde{\pi}_2$ are fixed is still non-negligible.

We split now the proof into two sub-cases based on the schedule of the messages of MIM: schedule of type (1) in which MIM plays Round 7 (i.e., message $\tilde{c}_1$) after the commitment phase is terminated in the left session; schedule of type (2) in which MIM plays Round 7 (i.e., message $\tilde{c}_1$) before the commitment phase is terminated in the left session. Notice that when the event Bad happens the schedule must be either of type (1) or type (2).

We proceed now proving that for both types of schedules, the hypothesis that Bad verifies with non-negligible probability leads to a contradiction.

Case 1: all schedules of type (1). If Bad happens with non-negligible probability $\widetilde{p}$ with schedules of type (1) our first goal is to show a successful man-in-the-middle adversary $\mathsf{MIM}_{\mathsf{5Ext}}$ and a corresponding distinguisher $\mathsf{D}_{\mathsf{5Ext}}$ for Π_{5Ext}. Then we will use them to reach a contradiction to the hiding of Π_{5Ext}.

More in detail, we define the man-in-the-middle $\mathsf{MIM}_{\mathsf{5Ext}}$ which plays as a receiver in the left session interacting with an honest sender $\mathcal{C}_{\mathsf{5Ext}}$ committing either to[10] a message m_0 or to a message m_1 and as a sender in the right session of Π_{5Ext} against an honest receiver $\mathcal{R}_{\mathsf{5Ext}}$.

When playing as a receiver, therefore getting messages from $\mathcal{C}_{\mathsf{5Ext}}$, in the left session of Π_{5Ext}, $\mathsf{MIM}_{\mathsf{5Ext}}$ internally runs MIM in a right session of Π_{NM} where MIM plays the role of a prover and $\mathsf{MIM}_{\mathsf{5Ext}}$ plays the role of a verifier. Therefore in the right session of Π_{NM}, $\mathsf{MIM}_{\mathsf{5Ext}}$ playing the subprotocol Π_{5Ext} as a sender will forward the messages of $\mathcal{C}_{\mathsf{5Ext}}$ to MIM. In the internal execution of Π_{NM} we have a left session where MIM will be a sender in the subprotocol Π_{5Ext}, and $\mathsf{MIM}_{\mathsf{5Ext}}$ will be a receiver. The messages of the subprotocol Π_{5Ext} that $\mathsf{MIM}_{\mathsf{5Ext}}$ will receive in this left session of the internal execution of Π_{NM} will be forwarded to the honest receiver $\mathcal{R}_{\mathsf{5Ext}}$ playing in the right session of Π_{5Ext}. Similarly, there will be a symmetric flow of messages in the opposite direction that for completeness we report now. Messages sent by $\mathcal{R}_{\mathsf{5Ext}}$ played in the right execution of Π_{5Ext} will be forwarded by $\mathsf{MIM}_{\mathsf{5Ext}}$ to MIM in the left session of the internal execution of Π_{NM} where indeed MIM is a sender of the subprotocol Π_{5Ext}. In turn, MIM in the right session of the internal execution of Π_{NM} will play messages as a receiver of the subprotocol Π_{5Ext} that $\mathsf{MIM}_{\mathsf{5Ext}}$ will receive and forward to $\mathcal{C}_{\mathsf{5Ext}}$ in the left session of Π_{5Ext}. $\mathsf{MIM}_{\mathsf{5Ext}}$ will play π_1 in the left session of the internal execution of Π_{NM} precisely as done in $\mathcal{H}_3$. Upon finishing the commitment phases of the subprotocol Π_{5Ext} in the left (resp., right) session of Π_{NM} (i.e., in the right (resp., left) session of Π_{5Ext}) with MIM, $\mathsf{MIM}_{\mathsf{5Ext}}$ sets $\tau_{\mathsf{5Ext}} = (com_1, com_2, com_3, com_4, com_5)$ (resp., $\widetilde{\tau}_{\mathsf{5Ext}} = (\widetilde{com}_1, \widetilde{com}_2, \widetilde{com}_3, \widetilde{com}_4, \widetilde{com}_5)$). Then $\mathsf{MIM}_{\mathsf{5Ext}}$ terminates giving in output $\mathsf{view}_{\mathsf{5Ext}} = (m, \widetilde{\tau}_{\mathsf{5Ext}}, \tau_{\mathsf{5Ext}}, r^*)$ where r^* is its randomness.

Notice that, by definition, $\mathsf{D}_{\mathsf{5Ext}}$ runs on input the message $\tilde{m}$ committed by $\mathsf{MIM}_{\mathsf{5Ext}}$ (i.e., the commitment generated by MIM while acting as a sender in the subprotocol Π_{5Ext} in the left session of Π_{NM}). Moreover, $\mathsf{D}_{\mathsf{5Ext}}$ takes as an input the view given in output by $\mathsf{MIM}_{\mathsf{5Ext}}$ that includes the randomness r^* used by $\mathsf{MIM}_{\mathsf{5Ext}}$ in the above-described execution of Π_{NM}. Therefore, the distinguisher $\mathsf{D}_{\mathsf{5Ext}}$ that we describe now (recall that our goal is to show a successful pair $(\mathsf{MIM}_{\mathsf{5Ext}}, \mathsf{D}_{\mathsf{5Ext}})$ in the indistinguishability game of the non-malleable commitment definition) interacts internally with MIM, resumes the execution of Π_{NM} described above, and then computes and sends Round 7 of the left session of Π_{NM} to MIM, namely $c_1 = \pi^2 \oplus \tilde{m}$. $\mathsf{D}_{\mathsf{5Ext}}$ upon receiving Round 7 of the right session from MIM (namely, $\tilde{c}_1$) does the following: if $\widetilde{\pi^2} = m_b \oplus \tilde{c}_1$ then $\mathsf{D}_{\mathsf{5Ext}}$ outputs b, otherwise $\mathsf{D}_{\mathsf{5Ext}}$ outputs a random bit, where $\widetilde{\pi^2}$ is the value that was already opened by MIM in the right session of the main thread. $\mathsf{MIM}_{\mathsf{5Ext}}$ and $\mathsf{D}_{\mathsf{5Ext}}$ are successful according to Definition 2.6.

Indeed the fact that the event Bad happens with non-negligible probability implies that a run of the above execution of $\mathsf{MIM}_{\mathsf{5Ext}}$ when the honest sender commits to m_b leads with non-negligible probability to $\mathsf{D}_{\mathsf{5Ext}}$ giving in output 1 in addition to flipping the coin in the other cases (i.e., when the event Bad does not happen). This means that the output of $\mathsf{D}_{\mathsf{5Ext}}$ is 1 with a probability non-negligibly larger than $1/2$. Instead, when the honest sender does not commit to m_b we have that $\mathsf{D}_{\mathsf{5Ext}}$ outputs 1 with probability non-larger than $1/2$ plus some negligible function.

The existence of the above $\mathsf{MIM}_{\mathsf{5Ext}}$ therefore implies, as guaranteed by Theorem 5.3 that there exists a special and partial extractor $\mathsf{Ext}_{\mathsf{5Ext}}$ that given in input the transcript of the commitment phase of Π_{5Ext}

[10] Recall that the distribution of the message committed by a MiM is independent of the message committed by the sender when the commitment scheme is non-malleable.

generated by $\mathsf{MIM}_{\mathsf{5Ext}}$ (and with oracle access to $\mathsf{MIM}_{\mathsf{5Ext}}$) extracts with non-negligible probability[11] the message committed by $\mathsf{MIM}_{\mathsf{5Ext}}$ in the right session of Π_{5Ext}, which in turn corresponds to the message committed by MIM in the left session of the execution of Π_{NM} that is run internally by $\mathsf{MIM}_{\mathsf{5Ext}}$.

We now describe the adversary $\mathcal{A}_{\mathsf{ExpHiding}}$ that breaks the hiding of Π_{5Ext} interacting with a challenger $\mathsf{C}_{\mathsf{ExpHiding}}$.

The reduction $\mathcal{A}_{\mathsf{ExpHiding}}$ internally uses $\mathsf{MIM}_{\mathsf{5Ext}}$, MIM and $\mathsf{Ext}_{\mathsf{5Ext}}$ providing them the needed randomness. Moreover, $\mathcal{A}_{\mathsf{ExpHiding}}$ has hard-coded the value $\widetilde{\pi^2}$ that is the value appeared in the main thread in the right session when π^2 is forced on the left session. As such, $\mathcal{A}_{\mathsf{ExpHiding}}$ can recompute all messages of the execution of Π_{NM} that is played internally by $\mathsf{MIM}_{\mathsf{5Ext}}$ with MIM. $\mathcal{A}_{\mathsf{ExpHiding}}$ works again considering two sessions, we will refer to them following the places where such commitments are played in the sessions of Π_{NM}; the right session will see the challenger $\mathsf{C}_{\mathsf{ExpHiding}}$ playing the role of $\mathcal{C}_{\mathsf{5Ext}}$, while the left session is the one where $\mathsf{MIM}_{\mathsf{5Ext}}$ tries to commit to a related value that will be extracted using $\mathsf{Ext}_{\mathsf{5Ext}}$ without rewinding $\mathsf{C}_{\mathsf{ExpHiding}}$. The reduction works as follows:

1. $\mathcal{A}_{\mathsf{ExpHiding}}$ chooses two messages $\tilde{c}_0^0$ and $\tilde{c}_0^1$ sampled at random from $\{0,1\}^\lambda$ and sends them to $\mathsf{C}_{\mathsf{ExpHiding}}$. $\mathsf{C}_{\mathsf{ExpHiding}}$ computes $\widetilde{com}_1$ w.r.t. $\tilde{c}_0^b$ for some randomly chosen bit b and sends it to $\mathcal{A}_{\mathsf{ExpHiding}}$. $\mathcal{A}_{\mathsf{ExpHiding}}$ obtains $\widetilde{com}_1$ from $\mathsf{C}_{\mathsf{ExpHiding}}$ and sends it to $\mathsf{MIM}_{\mathsf{5Ext}}$.

2. Upon receiving com_i from $\mathsf{MIM}_{\mathsf{5Ext}}$ in the left execution $\mathcal{A}_{\mathsf{ExpHiding}}$ computes com_{i+1} as an honest receiver and send it to $\mathsf{MIM}_{\mathsf{5Ext}}$, for $i \in \{1,3\}$.

3. Upon receiving $\widetilde{com}_j$ from $\mathsf{MIM}_{\mathsf{5Ext}}$ in the right execution $\mathcal{A}_{\mathsf{ExpHiding}}$ asks for $\widetilde{com}_{j+1}$ to $\mathsf{C}_{\mathsf{ExpHiding}}$ and then $\mathcal{A}_{\mathsf{ExpHiding}}$ forwards it to MIM, for $j \in \{2,4\}$.

4. Upon finishing the commitment phase of Π_{5Ext} in the left session, $\mathcal{A}_{\mathsf{ExpHiding}}$ runs $\mathsf{Ext}_{\mathsf{5Ext}}$ on input the transcript $\tau = (com_1, \ldots, com_5)$ generated in this session, and $\mathsf{Ext}_{\mathsf{5Ext}}$ will get oracle access to $\mathsf{MIM}_{\mathsf{5Ext}}$.

   At the end of this phase $\mathsf{Ext}_{\mathsf{5Ext}}$ outputs c_0^* (recall that with non-negligible probability it corresponds to the value committed by $\mathsf{MIM}_{\mathsf{5Ext}}$ which in turn corresponds to the share for the challenge of Π played by MIM in the left execution of Π_{NM}).

5. $\mathcal{A}_{\mathsf{ExpHiding}}$ continues the execution of the left and right sessions of Π_{NM} played internally by $\mathsf{MIM}_{\mathsf{5Ext}}$ with MIM and that were interrupted when τ was obtained; $\mathcal{A}_{\mathsf{ExpHiding}}$ will use c_0^* as described in $\mathcal{H}_3$ and will continue the execution of Π_{NM} until obtaining $\tilde{c}_1$ from MIM.

6. $\mathcal{A}_{\mathsf{ExpHiding}}$ checks if $\widetilde{\pi^2} = \tilde{c}_0^0 \oplus \tilde{c}_1$, and if so $\mathcal{A}_{\mathsf{ExpHiding}}$ outputs 0. If $\widetilde{\pi^2} = \tilde{c}_0^1 \oplus \tilde{c}_1$ then $\mathcal{A}_{\mathsf{ExpHiding}}$ outputs 1. Otherwise $\mathcal{A}_{\mathsf{ExpHiding}}$ outputs a random bit.

Note that if the value c_0^* is correct, then it corresponds to the value extracted (through the extractor of Π_{5Ext}) and then used in $\mathcal{H}_3$. Since $\mathsf{Ext}_{\mathsf{5Ext}}$ succeeds with non-negligible probability, we have that the above run of $\mathcal{A}_{\mathsf{ExpHiding}}$ with non-negligible probability corresponds to a run of $\mathcal{H}_3$. Since we know that in a run of $\mathcal{H}_3$ (by contradiction) the event Bad happens with non-negligible probability we have that the reduction with non-negligible probability will not output a random bit, since it will correctly guess b. At the same time, we observe that the probability that the reduction will not output a random bit and will instead output the incorrect bit is negligible since the value $\tilde{c}_0^{1-b}$ is sampled uniformly at random and is unconditionally hidden in the experiment.

From the above arguments we can conclude that $\mathcal{A}_{\mathsf{ExpHiding}}$ breaks the hiding of Π_{5Ext} with non-negligible probability and this contradicts the hypothesis that Bad happens with non-negligible probability in $\mathcal{H}_3$. Therefore in $\mathcal{H}_3$ the Extracting Stage succeeds in obtaining a witness with a probability that is negligibly close to the one of $\mathcal{H}_2$.

[11] We recall that in this part of the proof we are assuming that the event Bad happens with non-negligible probability for schedules of type (1), and $\mathsf{Ext}_{\mathsf{5Ext}}$ extracts the committed message, for schedules of that type, with non-negligible probability.

Case 2: all the schedules of type (2). In this type of schedule the event Bad occurs with negligible probability since otherwise we can again show a reduction to the hiding of Π_{5Ext}. The reduction proceeds exactly as described in Lemma 6.2 and terminates after obtaining Round 7 of the right session of Π_{NM} which, in this case, is scheduled *before* the end of the commitment phase in the left session[12]. Therefore there is not even a need to run an extraction on the left session and thus there is no issue with respect to the challenger of the hiding property that is instead acting as a sender of the subprotocol Π_{5Ext} played in the right session.

The above two cases cover all possible schedules and this observation concludes the proof of the indistinguishability of $\mathcal{H}_2$ and $\mathcal{H}_3$. □

Lemma 6.5. *$\mathcal{H}_4$ is computationally indistinguishable from $\mathcal{H}_3$.*

Proof. Assume by contradiction that the claim does not hold, therefore there exists a distinguisher D which distinguishes the view produced by MIM in $\mathcal{H}_3$ from the one produced in $\mathcal{H}_4$ with non-negligible probability. We now show an adversary $\mathcal{A}_{\mathsf{ExpZK}}$ for the SHVZK property of Π.

$\mathcal{A}_{\mathsf{ExpZK}}$ receives in input a transcript (π^1, π^2, π^3) from the challenger w.r.t. an instance $x \in \mathcal{L}$. Then, the reduction proceeds as follows. $\mathcal{A}_{\mathsf{ExpZK}}$ sends π^1 to MIM in the left execution of Π_{NM}. Further, $\mathcal{A}_{\mathsf{ExpZK}}$ continues the execution with MIM on the left execution of Π_{NM} and on the right execution of Π_{NM} following $\mathcal{H}_4$ until Round 7. Next, in Round 7 $\mathcal{A}_{\mathsf{ExpZK}}$ sets $c_1 = \pi^2 \oplus c_0^*$, where c_0^* is the value extracted in the left execution of Π_{NM} (specifically, c_0^* is obtained rewinding once MIM from Round 5 to Round 4 of Π_{5Ext} in the left session and sending to MIM a different message in Round 4 of Π_{5Ext}[13]). $\mathcal{A}_{\mathsf{ExpZK}}$ sends c_2 to MIM. $\mathcal{A}_{\mathsf{ExpZK}}$ continues the execution with MIM until Round 9 in both left and right sessions. Next, in Round 9 of the left execution of Π_{NM}, $\mathcal{A}_{\mathsf{ExpZK}}$ sends to MIM the value π^3 received in input. Notice that if (π^1, π^2, π^3) is computed by the SHVZK simulator of Π, then the execution corresponds exactly to $\mathcal{H}_4$. If instead (π^1, π^2, π^3) is computed by the prover of Π, then the execution corresponds exactly to $\mathcal{H}_3$. Therefore $\mathcal{A}_{\mathsf{ExpZK}}$ runs the distinguisher D and breaks with non-negligible probability the SHVZK of Π.

The event Bad happens in $\mathcal{H}_4$ with negligible probability, otherwise, observe that a run of $\mathcal{H}_4$ would lead with non-negligible probability to the same $\widetilde{\pi}^2$ appearing both in the first output of the experiment, and in the Extraction Stage. As proven in Lemma 6.4, the above event happens only with negligible probability in $\mathcal{H}_3$. Therefore one can again show a reduction to the SHVZK of Π that follows the one just shown, except that the reduction will succeed by running the experiment until $\widetilde{\pi}^2$ is played during the Extraction Stage. Clearly the reduction will have an advantage in distinguishing between the transcript of a SHVZK simulator of Π and the one of a prover of Π. Being this reduction to the SHVZK of Π very similar to the previous one, we omit further details.

Finally, observe that both the proven indistinguishability of the transcripts and the negligible probability that the event Bad happens apply to any schedule of the left and right sessions. The observations that the distributions of the views of MIM in $\mathcal{H}_3$ and $\mathcal{H}_4$ are computationally indistinguishable, and that the probability of extracting a witness in $\mathcal{H}_4$ is negligibly close to the one in $\mathcal{H}_3$, conclude the proof. □

$\mathcal{H}_5$ is equal to $\mathsf{Sim}_{\mathsf{NM}}$ that therefore produces a transcript that is computationally indistinguishable from a real transcript. Moreover, $\mathsf{Sim}_{\mathsf{NM}}$ fails only with negligible probability in giving in output also a witness w^* for the accepting right session appearing the right session of the transcript.

Running time. The running time of $\mathsf{Sim}_{\mathsf{NM}}$ consists of a run of the extractor of Π_{5Ext} on the left session and a run of the extractor of Π in the right session to get a witness. Note that the two extraction procedures are independent and in particular, the new messages played in the Extracting Stage are identically distributed to the ones of an honest verifier. Therefore, even in case the scheduling of the messages is such that rewinds to get a witness on the right session require each time to simulate from scratch the left session (and thus to extract again from the extractable commitment in the left session), the overall expected running time remains polynomial (as discussed in [KL05]).

[12] Note that the reduction stops before sending Round 7 in the left session, therefore it does need to extract the message committed by MIM in the left session.

[13] Note that since MIM is non-aborting with non-negligible probability and Π_{5Ext} satisfies Lemma 5.1, then $c_0^* \neq \perp$ is extracted with non-negligible probability which is sufficient for terminating the reduction.

□

Corollary 6.5.1. *Assuming the existence of OWFs (resp., 1-1 OWFs), there exists a 10-round (resp., 9-round) NMZK, which makes black-box use of OWFs (resp., 1-1 OWFs).*

The corollary holds since Π_{5Ext} only uses OWFs (resp., 1-1 OWFs) in a black-box fashion and Π can be instantiated using [IKOS07] which also uses OWFs (resp., 1-1 OWFs) in a black-box way.

The above corollary differentiates the two cases in terms of round complexity because the 1st round of Π could need to compute a non-interactive commitment and this requires a preliminary round in order to be instantiated with OWFs.

Acknowledgements

The first author received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under project PROCONTRA (grant agreement No. 885666) and by the project PARTHENON (B53D23013000006), under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

The last author is member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM) and his research contribution on this work is financially supported under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union - NextGenerationEU - Project Title "PARTHENON" - CUP D53D23008610006 - Grant Assignment Decree No. 959 adopted on June 30, 2023 by the Italian Ministry of Ministry of University and Research (MUR).

References

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, October / November 2017.

[BGR+15] Hai Brenner, Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. Fast non-malleable commitments. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 1048–1057. ACM Press, October 2015.

[BJY97] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 280–305. Springer, Heidelberg, May 1997.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1825–1842. ACM Press, October / November 2017.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, August 1994.

[COS22] Michele Ciampi, Emmanuela Orsini, and Luisa Siniscalchi. Four-round black-box non-malleable schemes from one-way permutations. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 300–329. Springer, Heidelberg, November 2022.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four-round concurrent non-malleable commitments from one-way functions. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 127–157. Springer, Heidelberg, August 2017.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pages 542–552. ACM Press, May 1991.

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd FOCS*, pages 51–60. IEEE Computer Society Press, October 2012.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. ZKBoo: Faster zero-knowledge for Boolean circuits. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 1069–1083. USENIX Association, August 2016.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001.

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 695–704. ACM Press, June 2011.

[GRRV14] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *55th FOCS*, pages 41–50. IEEE Computer Society Press, October 2014.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 201–215. Springer, Heidelberg, August 1996.

[HV18] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. Round-optimal fully black-box zero-knowledge arguments from one-way permutations. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 263–285. Springer, Heidelberg, November 2018.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[JP14] Abhishek Jain and Omkant Pandey. Non-malleable zero knowledge: Black-box constructions and definitional relationships. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 435–454. Springer, Heidelberg, September 2014.

[KL05] Jonathan Katz and Yehuda Lindell. Handling expected polynomial-time strategies in simulation-based security proofs. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 128–149. Springer, Heidelberg, February 2005.

[KLP22a] Allen Kim, Xiao Liang, and Omkant Pandey. A new approach to efficient non-malleable zero-knowledge. Cryptology ePrint Archive, Report 2022/767, 2022. `https://eprint.iacr.org/2022/767`.

[KLP22b] Allen Kim, Xiao Liang, and Omkant Pandey. A new approach to efficient non-malleable zero-knowledge. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 389–418. Springer, Heidelberg, August 2022.

[KOS18] Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. Round optimal black-box "commit-and-prove". In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 286–313. Springer, Heidelberg, November 2018.

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 128–136. Springer, Heidelberg, August 1990.

[PR05] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 533–542. ACM Press, May 2005.

[PR08] Rafael Pass and Alon Rosen. New and improved constructions of nonmalleable cryptographic protocols. *SIAM Journal on Computing*, 38(2):702–752, 2008.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 403–418. Springer, Heidelberg, March 2009.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51st FOCS*, pages 531–540. IEEE Computer Society Press, October 2010.