A New Approach to Efficient Non-Malleable Zero-Knowledge*

Allen Kim, Xiao Liang[0000−0003−0858−9289], and Omkant Pandey

Stony Brook University, Stony Brook, USA
allekim@cs.stonybrook.edu
xiao.crypto@gmail.com
omkant@cs.stonybrook.edu

Abstract. Non-malleable zero-knowledge, originally introduced in the context of man-in-the-middle attacks, serves as an important building block to protect against concurrent attacks where different protocols may coexist and interleave. While this primitive admits almost optimal constructions in the plain model, they are *several* orders of magnitude slower in practice than standalone zero-knowledge. This is in sharp contrast to non-malleable *commitments* where practical constructions (under the DDH assumption) have been known for a while.

We present a new approach for constructing efficient non-malleable zero-knowledge for all languages in $\mathcal{NP}$, based on a new primitive called *instance-based non-malleable commitment* (IB-NMC). We show how to construct practical IB-NMC by leveraging the fact that *simulators* of *sub-linear* zero-knowledge protocols can be much faster than the honest prover algorithm. With an efficient implementation of IB-NMC, our approach yields the first general-purpose non-malleable zero-knowledge protocol that achieves practical efficiency *in the plain model*.

All of our protocols can be instantiated from symmetric primitives such as block-ciphers and collision-resistant hash functions, have reasonable efficiency in practice, and are general-purpose. Our techniques also yield the first efficient non-malleable commitment scheme *without public-key assumptions*.

Keywords: Non-malleability · Efficiency · Symmetric Assumptions.

1 Introduction

Non-malleable Zero-Knowledge. Dolev, Dwork, and Naor [27] introduced the notion on non-malleable cryptography. They also provided constructions of non-malleable zero-knowledge and non-malleable commitments in the *plain*

* This material is based upon work supported in part by DARPA SIEVE Award HR00112020026, NSF CAREER Award 2144303, NSF grants 1907908, 2028920, 2106263, and 2128187. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA, or NSF.

model assuming only the existence of *one-way functions* (OWFs). While these primitives were originally introduced in the context of "man-in-the-middle" attacks, they were soon used as a *building block* for constructing secure computation protocols. For example, non-malleable commitments were used extensively to improve their round-efficiency [49, 69, 81, 41, 33, 3, 20], and non-malleable zero-knowledge played a central role in protecting them against concurrent attacks [13, 14, 19, 71, 76, 62, 12].

A long line of research has since focused on several aspects of these primitives, including their round-complexity [27, 5, 74, 60, 55, 81, 57, 41, 21, 50], black-box usage of underlying primitives [42, 45], and even concrete efficiency [11] without assuming any trusted setup. Notably, constant-round non-malleable commitments assuming only OWFs were first constructed in independent and concurrent works of Goyal [41] and Lin and Pass [57]. Finally, four-round non-malleable *zero-knowledge* assuming only OWFs was first achieved by Goyal et al. [45] for all of $\mathcal{NP}$; and three-round non-malleable commitments assuming injective OWFs were constructed by Goyal, Pandey, and Richelson [43, 44]. Under falsifiable assumptions [65, 34], these rounds are optimal for commitments [72], and likely to be optimal for zero-knowledge as well [39, 32]. Stronger forms of this notion such as *concurrent* non-malleability, eventually achieved optimally in a series of works [73, 60, 21], are not considered in this work. We note that non-malleability has been explored in several other contexts as well [27, 10, 26, 30].

Efficient Constructions. While the aforementioned results are almost optimal for non-malleable zero-knowledge, their focus is primarily on *feasibility* as opposed to actual efficiency. To the best of our knowledge, the actual efficiency of non-malleable zero-knowledge has never been explicitly addressed before. This is in sharp contrast to non-malleable *commitments*, for which efficient plain-model constructions are known (under the DDH assumption) [11].

We therefore consider the efficiency of some of the main approaches for non-malleable zero-knowledge. Unless stated otherwise, we are concerned with *general-purpose* protocols (that work for all languages in $\mathcal{NP}$) *in the plain model*.

- The most common approach for non-malleable zero-knowledge is "commit-and-prove." At a high level, the prover first sends a *non-malleable* commitment to the witness, and then uses (ordinary) zero-knowledge to prove that the committed value is a valid witness [27, 7, 45, 21]. If the commitment supports k-bit identities and has λ-bit security, the circuit corresponding to the state-of-the-art non-malleable commitment [11] is at least $16k^2\lambda^2$, or over 100 million gates for $k = 32, \lambda = 80$. Zero-knowledge proofs for such circuits would take more than one minute using state-of-the-art (plain-model) protocols such as Ligero [2] (even taking advantage of the amortization admitted by Ligero). This is true even if the actual statement, say proving $y = \mathrm{SHA256}(x)$, requires less than a second [2] in the standalone case.[1]

[1] Although details may vary, known protocols in this paradigm generally require some form of non-algebraic consistency proof over a non-malleable commitment supporting large identities and message spaces.

It is worth noting that using state-of-the-art commitments [11] additionally requires assuming DDH, whereas "symmetric assumptions" such as OWFs are sufficient in theory. Efficient non-malleable commitments *without* relying on public-key assumptions such as DDH are therefore also not known. One option here is to implement the consistency proofs in [11] with Ligero to avoid DDH. However, this also results in large circuits.[2] Jumping ahead, our techniques offer new results for efficient non-malleable commitments, too.

– Non-malleable zero-knowledge *without* relying on non-malleable commitments was first constructed by Barak [5], and by Pass and Rosen [74] under improved assumptions. Both of these constructions were based on Barak's non-black-box simulation [4]. A critical component of these protocols is a universal argument [6], which consists of a Merkle tree commitment to a *Probabilistically Checkable Proof* (PCP), parts of which are opened later in the protocol. Unfortunately, as shown by Ben-Sasson et al. [9], the underlying PCP proof in the universal argument can be astronomically large even for moderate parameters. To the best of our knowledge, the true efficiency of non-black-box simulation based constructions is currently not well understood.

– A third approach, due to Ostrovsky, Pandey, and Visconti [68], relies on the DDH assumption, and efficiently converts any public-coin honest-verifier statistical zero-knowledge argument into a (concurrent) non-malleable one [7]. While this approach uses non-malleable commitments, it avoids general-purpose proofs over them using ideas from the "simulatable commitment" of Micciancio and Petrank [63]. Though efficient, this transformation quickly becomes pretty slow. For example, for the standalone setting, it requires roughly $20k\lambda \log \lambda$ group exponentiations to support k bit identities at $2^{-\lambda}$ security level;[3] this is roughly 0.32 million exponentiations for $k = 32, \lambda = 80$. In addition, it requires efficient non-malleable commitments as well as efficient (and compatible) simulatable commitments, both of which are only known from DDH. Ideally, we would like to use only symmetric assumptions.

Constructions in the Random Oracle Model (ROM). The protocols we seek are straightforward to construct in the ROM [8] (see, e.g., [70, 31]). Briefly, a random oracle (RO) is non-malleable by design, which completely sidesteps this issue. Furthermore, zero-knowledge is also trivial since the simulator and

[2] We remark that for non-malleable commitments based on non-malleable codes such as [43], it is hard to estimate the overall complexity; the asymptotic analysis of underlying codes such as [1] has astronomically large constants, making them unsuitable in practice.

[3] The analysis in [68] does not separate identity lengths from security levels; it further provides only asymptotic analysis which hides multiplicative constants and does not specify the exact negligible and super-logarithmic functions. This makes it difficult to assess the security level supported by their protocol. If the analysis is performed to support λ-bit security and k-bit identities, the overhead is at least $20k\lambda \log \lambda$ group exponentiations.

the reduction are allowed to see adversary's queries to the oracle and control the responses. In the real world, a cryptographic hash function is used to replace the RO, thus providing a concrete construction. This is an attractive methodology that often leads to practical constructions. That being so, there are several reasons to pursue constructions in the plain model, *even if efficient constructions are already known in the ROM*. We highlight some of them here.

- A protocol such as a zero-knowledge proof in the ROM can be particularly troublesome when it is used as a sub-protocol in a larger protocol. If the RO is shared by other parts of the larger protocol, the security is jeopardized since the security reduction for the sub-protocol does not hold when a particular RO has already been selected by the larger protocol (see, e.g., [70, 15, 79]). In addition, security proofs in this model often *program* the oracle, resulting in loss of properties such as *deniability* which are otherwise implied by zero-knowledge (see [70, 80]). Deniability is a natural and useful property that has been explored in other contexts as well [16, 29, 67, 78].
- Using random oracles often sidesteps the main difficulty in achieving a particular task, such as CCA secure encryption or non-malleable commitments from standard assumption. Therefore, a construction or security proof in the ROM, while valuable, is usually not as *insightful* as its plain-model counterparts.
- Finally, while security proofs in the ROM are valuable, it requires a leap of faith to believe that instantiating the random oracle with a real world hash function maintains claimed security. Indeed, this is not always the case [17, 28, 66, 40]. It stands to reason that whenever possible the ROM should be avoided.

Improved constructions can be achieved in other trusted setup models as well. Di Crescenzo, Ishai, and Ostrovsky [24] construct non-interactive non-malleable commitments in the CRS model, and Di Crescenzo et al. [25] do so efficiently under DDH. Lower rounds can also be achieved in the plain model under non-falsifiable assumptions [69, 72, 58, 51].

1.1 Our Results

We present a new approach for constructing *efficient* and *general-purpose* non-malleable zero-knowledge *in the plain model*. Our protocols can be viewed as a transformation which takes as input an efficient general-purpose zero-knowledge protocol, such as Ligero [2], and yields a *non-malleable* zero-knowledge protocol of (less but still) comparable efficiency. To the best of our knowledge, this is the first construction of general-purpose non-malleable zero-knowledge that achieves practical efficiency in the plain model. Our approach has the additional benefit of requiring only *symmetric* assumptions (in addition to the assumptions of the given proof system). Specifically, it suffices to assume collision-resistant hash functions.

Table 1: Performance of our protocols for λ-bit security and k-bit identities. NMZK proves a witness for SHA256.

Param.	**NMZK**			**NMCom**		
(k, λ)	P time (s)	V time (s)	Comm. (MB)	P time (s)	V time (s)	Comm. (MB)
(32, 40)	1.68	0.74	19.68	2.52	1.12	19.74
(32, 80)	3.56	1.49	24.88	4.68	2.06	24.97
(64, 80)	5.04	2.23	28.84	6.72	3.09	28.93

While our primary focus is on non-malleable zero-knowledge, we also get new results for non-malleable *commitments*. Specifically, we get the first efficient construction of non-malleable commitments with large identities and message space *under symmetric assumptions*. Though this improves upon the DDH assumption required by the state-of-the-art construction [11], our construction is somewhat slower in comparison.

Even though our focus is on efficiency, our results are theoretical in nature. Our transformation makes use of non-malleable commitments in a fundamentally new way. We define and construct a new primitive called *instance-based non-malleable commitments* (IB-NMC), which admit more efficient modes than a traditional non-malleable commitment. We show how IB-NMC can be used in conjunction with the *OR-Composition* technique from [22, 23] to obtain efficient *simulation-sound* protocols, which in turn yields efficient non-malleable protocols for both zero-knowledge and commitments. This primitive may be useful in other contexts as well.

The overhead of our transformation is within reach of practical computing. Table 1 shows the running times and communication for our non-malleable protocols for some sample parameters. Due to space constraints, a detailed analysis of the empirical results is postponed to the full version [53].

1.2 Overview of Techniques

We start by recalling the central efficiency bottleneck in constructing non-malleable zero-knowledge for $\mathcal{NP}$. We assume that efficient *standalone* zero-knowledge (ZK) proofs already exist for all languages $L \in \mathcal{NP}$ in the plain model such as [35, 2]. For concreteness, we will use Ligero [2].

The main inefficiency of non-malleable zero-knowledge stems from the fact that almost all known constructions [7, 59, 56] make a non-black-box use of non-malleable commitments. More specifically, the prover commits to a witness or a trapdoor string using a non-malleable commitment and later relies on expensive $\mathcal{NP}$ reductions to prove that it either committed a valid witness *or* a trapdoor (i.e., an OR-statement); the latter is shown difficult to do for the man-in-the-middle adversary M by relying on the non-malleability of the commitment. The $\mathcal{NP}$ reduction corresponding to the OR-statement typically results in a circuit

description of formidable size since the non-malleable commitment usually contains many calls to cryptographic functions such as block-ciphers. The resulting protocols are prohibitively inefficient even with state-of-the-art ZK constructions. Other approaches (based on non-black-box simulation or DDH outlined earlier) are irrelevant to our construction.

The starting point of our work is the observation that the use of non-malleable commitments in these protocols is merely *a means to an end*. In particular, the honest prover generally commits to a random or an all-zero string in these commitments; it is the simulator who makes real use of their non-malleable properties. Therefore, if we can create a situation in our protocols where the honest prover *does not have to execute even a single full non-malleable commitment*, we can improve the computational efficiency of these protocols. Let us briefly highlight why achieving this property is extremely important for our goals: As noted above, efficient non-malleable commitments in the plain model are based on DDH [68, 11]. One option to avoid public-key assumptions is to instantiate the scheme in [11] with Ligero; However, the running time of the resulting commitment scheme *alone* (under moderate parameters) will run in more than one minute. The actual non-malleable zero-knowledge protocol which depends on these commitments in a non-black-box way will be much worse. We therefore seek to avoid even *one* full execution of a non-malleable commitment in our ZK protocol.

It is worthwhile to note that *black-box* constructions of non-malleable ZK from non-malleable commitments are (surprisingly) not known. The closest work in this regard is by Jain and Pandey [48], who construct simulation-sound ZK from a stronger version of non-malleable commitments (called *1-1 CCA* [18, 54]) in black-box. Currently, it is unclear if their approach can yield an efficient protocol that avoids even one execution of the non-malleable commitment.

Instance-Based Non-Malleable Commitments. Returning back to our goal of avoiding even one execution of full non-malleable commitment during the proof, we consider a new relaxation of such commitments which we call *instance-based* non-malleable commitments (IB-NMCs). Roughly speaking, an IB-NMC is just like an ordinary non-malleable commitment except that it takes as input a statement y (from an implicit $\mathcal{NP}$ language Y). The commitment has two modes: if $y \notin Y$, then it is an ordinary non-malleable commitment, and the committer commits to any desired value v by following the actual commitment algorithm C. Otherwise, if $y \in Y$, then the commitment is not guaranteed to have any non-malleability property. However, in this case, there exists a much faster algorithm C^* that, with the help of a witness for $y \in Y$, can *fake* (or simulate) an execution that looks indistinguishable from the real execution with C for any value v.

To construct IB-NMC, we combine the following key ideas:

– The simulator of a general-purpose zero-knowledge proof can be much faster than the real prover algorithm. This is best seen by considering the sub-linear zero-knowledge arguments based on PCPs [52, 61]. In such protocols,

a prover commits to a full Merkle tree over the PCP proof; but note that the simulator does not have to construct the whole tree. Instead, the simulator can simply prepare the nodes of the opened paths in a consistent manner, which is much faster. In particular, this is true for our chosen ZK system Ligero.

- The well-known OR-Composition technique developed for Σ-protocols [22, 23] can be applied in our setting to give proofs for statements of the type "either $x \in L$ or $y \in Y$." Recall that under this technique, a prover with a witness for x constructs proofs correctly for the "$x \in L$" part, but uses the *simulator* of the Σ-protocol for the "$y \in Y$" part. Observe that if the simulator for the "$y \in Y$" part is fast (as discussed in the previous item), then the composed proofs can be almost as fast as a proof *only* for $x \in L$.
- Finally, we apply the aforementioned observations to a suitable non-malleable commitment scheme to get an efficient IB-NMC. In particular, we apply it to a modification of the BGRRV protocol [11], leading to a construction based solely on symmetric-key (or Minicrypt) assumptions (referred to as $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$). More specifically, $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ has a *commit phase* and a *proof phase* where the latter proves the "consistency" of the former. To get IB-NMC, we simply change the proof phase to prove that either the first phase is consistent or $y \in Y$ (where y is an additional input to the committer); this proof is done using the OR-composition of two Ligero protocols as described above.

We remark that this approach runs into several other issues that are not discussed here, e.g., OR-composition in general applies only to Σ-protocols but Ligero is not a Σ-protocol, the role of Y and how to choose it, etc. We will handle them in Sec. 5. The use of a honest-verifier simulator to protect against malicious attacks first appears in the work of Cramer, Damgård, and Schoenmakers [22].

Non-Malleability via Simulation Soundness. While IB-NMC is an interesting primitive, it is not clear how to use it at all to construct non-malleable zero knowledge. Instead, we show that IB-NMC can be used successfully to construct a fast *simulation-sound* ZK protocol [77, 48]. Constructing this protocol requires repeated applications of the OR composition and the fake-proof technique discussed above. The simulation-sound protocol can be directly useful in larger protocols since this notion suffices for typical applications of non-malleability. Finally, we show how to use this protocol to get an efficient and full-fledged non-malleable ZK as well as an efficient non-malleable commitment. In both cases, the transformation inherits the assumptions of the underlying zero-knowledge and IB-NMC, which in our case, are symmetric primitives only.

2 Preliminaries

We use $\lambda \in \mathbb{N}$ to denote the security parameter. Symbols $\overset{\mathrm{c}}{\approx}$, $\overset{\mathrm{s}}{\approx}$, and $\overset{\mathrm{id}}{=}$ are used to denote computational, statistical, and perfect indistinguishability respectively. Let $\mathsf{negl}(\lambda)$ denote negligible functions. Familiarity with basic definitions including commitments, witness indistinguishability, zero-knowledge, arguments

of knowledge, etc. is assumed; we refer to [36, 37] for formal treatment of these notions. We also recall the definitions of CRHFs, extractable commitments, and statistically-hiding commitments in [53, Appendix A].

Non-Malleable Interactive Proofs. We work with identity-based (or "tag-based") definitions of non-malleability and follow the definitions and conventions from [74]. Let $\mathcal{A}$ be a (non-uniform) probabilistic Turing machine, specifying a man-in-the-middle strategy. $\mathcal{A}$ runs in time polynomial in the security parameter λ. Let $z \in \{0,1\}^*$ be an arbitrary string (denoting the non-uniform "advice" for $\mathcal{A}$). Let $\langle P, V, \rangle$ be an interactive proof system for an $\mathcal{NP}$ complete language L. Let $x \in L$ be a statement of length λ; we assume that P is PPT and receives a witness $w \in R_L(x)$ as its auxiliary input. The definition is based on the comparison between a *man-in-the-middle* execution and a *stand-alone* execution among the above parties.

The man-in-the-middle experiment begins by selecting uniform randomness for $\mathcal{A}$, and honest parties P and V. $\mathcal{A}(x,z)$ interacts with $P(x,w)$ on left acting as a verifier in the proof for $x \in L$; $\mathcal{A}$ simultaneously participates in a right proof with V, proving a related statement $\widetilde{x}$, supposedly in L.[4] Let the tag (or "identity") strings on left and right be id and $\widetilde{\mathsf{id}}$ respectively with $|\mathsf{id}| = |\widetilde{\mathsf{id}}| = \lambda$. We let $\mathsf{mim}_V^{\mathcal{A}}(\mathsf{id}, \widetilde{\mathsf{id}}, x, \widetilde{x}, w, z)$ be a random variable describing the output of V in the man-in-the-middle execution.

In the stand-alone execution, a machine S interacts with the honest verifier V. As in the man-in-the-middle execution, V receives as input an instance $\widetilde{x}$ and the identity $\widetilde{\mathsf{id}}$. S receives x, an auxiliary input z and id as input. We let $\mathsf{sta}_V^S(\mathsf{id}, \widetilde{\mathsf{id}}, x, \widetilde{x}, z)$ be a random variable describing the output of V in the stand-alone execution.

Definition 1 (Non-Malleable Interactive Proof). *An interactive proof $\langle P, V\rangle$ for language L is said to be non-malleable w.r.t. tags of length m if for every* PPT *man-in-the-middle adversary $\mathcal{A}$, there exists a* PPT *stand-alone prover S and a negligible function negl such that for every $x \in L$, every $w \in R_L(x)$, every $\widetilde{x} \in \{0,1\}^{|x|}$, every $\mathsf{id}, \widetilde{\mathsf{id}} \in \{0,1\}^m$ so that $\mathsf{id} \neq \widetilde{\mathsf{id}}$, and every $z \in \{0,1\}^*$, it holds that*

$$\Pr[\mathsf{mim}_V^{\mathcal{A}}(\mathsf{id}, \widetilde{\mathsf{id}}, x, \widetilde{x}, w, z) = 1] < \Pr[\mathsf{sta}_V^S(\mathsf{id}, \widetilde{\mathsf{id}}, x, \widetilde{x}, z) = 1] + \mathsf{negl}(|x|).$$

We will refer to *synchronizing* adversaries: they are the man-in-the-middle attackers who, upon receiving a message in one session, immediately respond with the corresponding message in the other session. An adversary is said to be *non-synchronizing* if it is not synchronizing.

Definition 2 (Non-Malleable Zero Knowledge). *An interactive proof between prover P and verifier V is said to be non-malleable zero knowledge if it is a non-malleable interactive proof that also has the zero-knowledge property.*

[4] We remark that statement $\widetilde{x}$ may be chosen either adaptively depending on the left execution, or statically by announcing it before the left execution begins.

Simulation Soundness. The notion of *simulation soundness* [77] is a form of non-malleable ZK. Typically it is all one needs when building higher-level constructs using non-malleable ZK. In the non-interactive setting, it requires that a man-in-the-middle adversary cannot generate convincing proofs for false statements, even given access to a simulator who can generate false proofs.

The definition for the interactive setting appears in [48]. It requires a *single* machine S—the simulator—which guarantees indistinguishability of the view for true statements (to capture ZK), and the soundness for statements on the *right hand side* even in the presence of simulated false proofs *on the left hand side*. We use $\mathsf{MIM}^{\mathcal{A}}_{\langle P,V\rangle}(x,w,z,\mathsf{id})$ to denote the joint view of the adversary $\mathcal{A}$ in the same man-in-the-middle execution described above.

Definition 3 (Simulation-Sound Zero-Knowledge). *An interactive argument $\langle P,V\rangle$ for a language L is said to be a* simulation-sound zero-knowledge argument *if for every* PPT *man-in-the-middle algorithm $\mathcal{A}$, there exists a expected PPT algorithm S (the simulator) such that:*

- **(Indistinguishable Simulation)** *For every $x \in L$, every $w \in R_L(x)$, every $\mathsf{id} \in \{0,1\}^\lambda$, and every (auxiliary input) $z \in \{0,1\}^*$:*

$$S(x,z,\mathsf{id}) \quad \overset{c}{\approx} \quad \mathsf{MIM}^{\mathcal{A}}_{\langle P,V\rangle}(x,w,z,\mathsf{id})$$

- **(Simulation Soundness)** *There exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $x \in \{0,1\}^\lambda$, every $\mathsf{id} \in \{0,1\}^\lambda$, and every $z \in \{0,1\}^*$:*

$$\Pr\left[\nu \leftarrow S(x,z,\mathsf{id}) : \widetilde{x} \notin L \wedge \widetilde{\mathsf{id}} \neq \mathsf{id} \wedge \widetilde{b} = 1\right] \leq \mathsf{negl}(\lambda)$$

  where $\widetilde{x}, \widetilde{\mathsf{id}}$ and $\widetilde{b}$ denote the statement, identity, and verifier's decision in the right-side view of the simulated joint-view ν.

Non-Malleable Commitments. We use the tag-based definition from [60, 43]. Specifically, we compare an ideal interaction with a real one. In the ideal interaction, a man-in-the-middle adversary $\mathcal{A}$ interacting with a committer C in the *left* session, and a receiver R in the *right*. We denote the relevant entities used in the right interaction as "tilde'd" version of the corresponding entities on the left. In particular, suppose that C commits to v in the left interaction, and $\mathcal{A}$ commits to $\widetilde{v}$ on the right. Let MIM_v denote the random variable that is the pair $(\mathsf{View}, \widetilde{v})$, consisting of the adversary's entire view of the man-in-the-middle execution as well as the value committed to by $\mathcal{A}$ on the right (assuming C commits to v on the left). The *ideal* interaction is similar, except that C commits to some arbitrary fixed value (say $0^{|v|}$, i.e. an all-zero string of length $|v|$) on the left. Let MIM_0 denote the pair $(\mathsf{View}, \widetilde{v})$ in the ideal interaction. We ensure that $\mathcal{A}$ uses a distinct identity (or "tag") $\widetilde{\mathsf{id}}$ on the right from the identity id it uses on the left. This is done by stipulating that MIM_v and MIM_0 both output a special value $\perp_{id}$ when $\mathcal{A}$ uses the same identity in both the left and right executions. Let $\mathsf{MIM}_v(z)$ and $\mathsf{MIM}_0(z)$ denote real and ideal interactions resp., when $\mathcal{A}$'s auxiliary input is z.

Definition 4 (Non-Malleable Commitments). *A tag-based statistically binding commitment scheme* $\langle C, R \rangle$ *is* non-malleable *if* $\forall$ *PPT* $\mathcal{A}$*, and* $\forall v \in \{0,1\}^\lambda$*, it holds that* $\{\mathsf{MIM}_v(z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*} \overset{c}{\approx} \{\mathsf{MIM}_0(z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$.

3 Preparatory Work

In this section, we prepare the ingredients for use in the construction of our non-malleable zero-knowledge protocol. More specifically, we recall how Ligero's ZK simulator works (from [2]). Also, we show a slightly-modified version of the non-malleable commitment from [11]. We will recall related notation/techniques only to the extent that is adequate to understand our construction. See the full version of the current paper for a more detailed review of Ligero [53].

On Notation. In [2], the authors first built a public-coin *zero-knowledge interactive PCP* (ZKIPCP) scheme. They then converted the ZKIPCP to a 6-round *honest-verifier* ZK protocol relying on Kilian's transformation [52, 61]. Finally, they further converted it to a 7-round (fully) zero-knowledge protocol using the techniques from [46, 47][5]. Henceforth, we will use Ligero to denote their *honest-verifier* ZK protocol, and use Ligero$'$ to denote their *fully* ZK construction.

Simulator HVSim for Ligero. We will use the fact that simulating a Ligero (i.e., the honest-verifier version of [2]) proof is *much faster* than the real prover algorithm if the challenge of the verifier is known. The simulator's algorithm will be denoted by HVSim (HV for "honest-verifier"). There are two parts to be simulated: the first one is simulating the ZKIPCP interaction (a.k.a. the challenge-response slot); and the second one is simulating paths of the Merkle tree that are consistent with opened parts of the ZKIPCP proof string π (a.k.a. the oracle query-answer slot). The full description of HVSim is presented in Algo. 1.

Algorithm 1: HVSim: Honest-Verifier Zero-Knowledge Simulator for Ligero

Input: a statement x, a collision-resistant hash function h, and a ZKIPCP query b:

1. Run the honest-verifier simulator algorithm corresponding to the ZKIPCP system for statement x and verifier randomness (h, b) to obtain a (perfectly) simulated ZKIPCP transcript. By definition, the transcript contains simulated parts of the "proof string" π. Let $L = \{(i, \pi_i)\}$ denote these simulated parts where $i \in [|\pi|]$ denotes position in the proof. Thus, L is simply the list of opened leaves in a Merkle tree (constructed below). Note that $n = |\pi|$ is the total number of leaves and known in advance. The simulated transcript also contains the honest verfier's challenge, which is simuilated as a random string b, and the corresponding (simulated) response c
2. Generate the paths of the Merkle tree that are consistent with $L = \{(i, \pi_i)\}$. This is straightforward, we provide the steps below for completeness:

[5] We remark that [2] also presented another approach—applying Fiat-Shamir transformation to their ZKIPCP will give a (fully) ZK protocol directly; moreover, the resulting protocol will be non-interactive. But this approach is irrelevant in the current paper as we are interested in constructions in the plain model (without random oracles).

(a) For every element $z = (i, \pi_i)$ in L, let i' represent the index corresponding to the sibling of z in the Merkle tree (note that i' exists for every i by definition). We first check if the sibling of z exists in L by checking if any element in L contains index i'. If no sibling for z exists in L, we add a new element $z' = (i', r_i)$ into L, where r_i is a random string with length equal to the output length of h.

(b) Let L^* be the empty set. For every z, along with its sibling z', in L, we let $z^* = h(z||z')$. We add z^* to L^*. In the end, the cardinality of L^* is equal to $|L|/2$.

(c) Set $L = L^*$ and $L^* = \emptyset$. Repeat Steps 2a to 2c while $|L| > 1$.

(d) The remaining element in L is the root of the Merkle tree.

Instantiating BGRRV with Symmetric Primitives. We will need an extractable non-malleable commitment (ENMC) that is fast and, preferably, based only on symmetric-key primitives. We work with a modified version of Brenner et al.'s protocol [11] (which is in turn based on [45]). This modified version uses Ligero' (the malicious-verifier version of Ligero) as the ZK proof system in the consistency-proof stage of the protocol. For concreteness, this instantiation is completely specified in Prot. 1. We refer to it as $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$.

Protocol 1: $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$: Extractable Non-Malleable Commitment in Minicrypt

Public Input: an identity $\mathsf{id} \in \{0,1\}^k$, a large prime q, an integer ℓ, and vector spaces $V_1, \ldots, V_n \subset \mathbb{Z}_q^\ell$ which are derived from id. These parameters satisfy the following relation: $\ell = 2(k+1)$ and $n = k+1$. (For the meanings of these parameters, we refer the readers to [11].)

Private Input: commiter C takes $\mathbf{m} \in \mathbb{Z}_q^{\ell-1}$ as its private input (the value to be committed).

Committing Stage. The committing stage consists of the following steps.

1. $R \to C$: Send the first message ρ of the Naor's commitment scheme [64].
2. $C \to R$: C chooses random values $r_1, \ldots, r_n \in \mathbb{Z}_q$. This defines vectors $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{Z}_q^\ell$ where $\mathbf{z}_i = (r_i, \mathbf{m})$. C sends commitments $(\widehat{\mathbf{m}}, \widehat{\mathbf{r}})$ where:

$$\widehat{\mathbf{m}} = \big(\mathsf{Com}_\rho(m_1; s_1), \ldots, \mathsf{Com}_\rho(m_{\ell-1}; s_{\ell-1})\big), \quad \widehat{\mathbf{r}} = \big(\mathsf{Com}_\rho(r_1; s'_1), \ldots, \mathsf{Com}_\rho(r_n; s'_n)\big),$$

   where Com_ρ denotes the second round of Naor's commitment w.r.t. first message ρ. Note that this commits C to every coordinate of $\mathbf{z}_i$. For future reference, define the following language which contains valid commitment and message pairs:

$$L_{\mathsf{Com}_\rho} := \big\{(c, a) : \exists b \text{ s.t. } c = \mathsf{Com}_\rho(a; b)\big\}.$$

3. $R \to C$: Send random challenge vectors $\{\mathbf{v}_i\}_{i=1,\ldots,n}$ where each $\mathbf{v}_i \in V_i \subset \mathbb{Z}_q^\ell$.
4. $C \to R$: C sends evaluations $\{w_i\}$, where each $w_i = \langle \mathbf{v}_i, \mathbf{z}_i \rangle \in \mathbb{Z}_q$.

Consistency Proof. Using Ligero', C proves that the preamble was executed correctly. That is, C proves the following statement: $\exists\big((m_1, s_1), \ldots, (m_{\ell-1}, s_{\ell-1}), (r_1, s'_1), \ldots, (r_n, s'_n)\big)$ such that

- $\widehat{\mathbf{m}} = \big(\mathsf{Com}_\rho(m_1; s_1), \ldots, \mathsf{Com}_\rho(m_{\ell-1}; s_{\ell-1})\big)$, **and**
- $\widehat{\mathbf{r}} = \big(\mathsf{Com}_\rho(r_1; s'_1), \ldots, \mathsf{Com}_\rho(r_n; s'_n)\big)$, **and**
- $w_i = \langle \mathbf{z}_i, \mathbf{v}_i \rangle \ \forall i \in [n]$ where $\mathbf{z}_i = (r_i, m_1, \ldots, m_{\ell-1})$.

Notation: Henceforth, we denote the above language as L^ρ_{consis}. We say that the above **Consistency Proof** stage is proving that $(\widehat{\mathbf{m}}, \widehat{\mathbf{r}}, \{w_i\}_{i\in[n]})$ is in language L^ρ_{consis}.

Observe that each message from the $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ receiver, informally speaking, is *efficiently simulatable* during "rewindings" given all prior information. That is, each message must be of one of the following three types: (i) it is a public random string; (ii) it can be sampled from scratch; *or* (iii) it is simply a complete opening of a previous commitment (and thus repeatable in rewind threads if needed). This observation will play an important role later when we prove the non-malleability of our ZK protocol (more specifically, in Claim 1). But we also emphasize that this observation is crucial *only* in the non-synchronous setting (but not in the synchronous setting).

Extractability of BGRRV. We remark that BGRRV is an extractable commitment scheme. Extraction can be performed from the preamble stage by simply rewinding to the second message, obtaining a valid answer for a different challenge, and then solving two equations in $\mathbb{Z}_p$.

4 Our Non-Malleable Zero Knowledge Protocol

In this section, we present the generic framework of our non-malleable zero-knowledge. Later in Sec. 5, we will instantiate each component of this protocol in special ways so that the final protocol admits an efficient implementation using only symmetric-key primitives. We use the following ingredients:

1. An extractable commitment scheme ExtCom. We will use the standard 3-round scheme from [75]. Note that the first committer message of this scheme is statistically-binding.

2. A tag-based commitment scheme ENMC that is both *non-malleable* and *extractable*; for concreteness, we will use scheme $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ specified in Prot. 1. We assume for convenience that the commitments are generated using Naor's scheme [64] w.r.t. an implicit first string ρ chosen by the receiver of the commitment (and dropped from the notation henceforth).

   We assume that the first *committer* message of ENMC is statistically binding. For concreteness, we say that a string c is an **honest** ENMC **commitment to a value** v **with tag** id if there exists randomness r such that c is the first committer message of ENMC produced by the honest committer algorithm on input value v, tag id, and randomness r.

3. A *statistically* witness-indistinguishable argument of knowledge sWIAoK.

Our construction is shown in Prot. 2 below. At a high level, the protocol is as follows: V starts by committing to a random string σ. P then uses an extractable non-malleable commitment ENMC to commit to an all-zero string. Then V decommits to its commitment made at the beginning of the protocol. Finally, P and V execute a sWIAoK protocol, where P proves to V that *either* it knows a witness to x, *or* that the commitment in ENMC equals σ.

Protocol 2: $\langle P, V\rangle_{\mathsf{NMZK}}$: Non-Malleable Zero-Knowledge

Public input: Security parameter λ, statement x (supposedly in an $\mathcal{NP}$ language L), and a tag $\mathsf{id} \in \{0,1\}^{\leq\lambda}$.

Private input: P takes the witness w as its private input.

1. V commits to a random string $\sigma \leftarrow \{0,1\}^\lambda$, using the extractable commitment scheme ExtCom. We denote the first committer message by com_1.
2. P commits to $\sigma' = 0^\lambda$ using the extractable non-malleable commitment ENMC with tag id. We denote the first committer message of this stage by com_2.
3. V sends σ along with decommitment information for com_1.
4. If Step 3 decommitment is valid, P proves the following compound statement to V using a *statistical* witness-indistinguishable argument of knowledge sWIAoK:
   - there exists a w such that $R(x,w) = 1$; **or**
   - com_2 is an honest ENMC commitment to σ with tag id.

For future reference (σ', r) is called the trapdoor witness for statement $(\mathsf{com}_2, \mathsf{id})$ if r is s.t. com_2 is the 1st committer message of ENMC on input σ', tag id, and randomness r.

Theorem 1. *The protocol $\langle P, V\rangle_{\mathsf{NMZK}}$ (shown in Prot. 2) is a non-malleable zero-knowledge argument of knowledge for $\mathcal{NP}$.*

To prove Thm. 1, we first need to prove that $\langle P, V\rangle_{\mathsf{NMZK}}$ is a zero-knowledge argument of knowledge. This follows from standard techniques. Due to space constraints, we postpone it to the full version [53]. In the following, we show the non-malleability of $\langle P, V\rangle_{\mathsf{NMZK}}$.

Lemma 1. *$\langle P, V\rangle_{\mathsf{NMZK}}$ is non-malleable.*

We prove Lem. 1 in subsequent subsections. We first present in Sec. 4.1 the proof regarding synchronous adversaries (who send their right messages as soon as they receive the corresponding left message). Then, we deal with the general case of non-synchronous adversaries in Sec. 4.2.

When reading the proofs in the synchronous setting, it would be helpful to keep in mind also the non-synchronous case. We add remarks at the end of each hybrid to address this. We hope it can improve the readability when we talk about the non-synchronous setting later.

4.1 Non-Malleability against Synchronous Adversaries

To prove non-malleability, we need to build a simulator which can convince V with roughly the same probability as a man-in-the-middle adversary $\mathcal{A}_{\mathsf{mim}}$ (up to some negligible difference), but without the help of the left interaction. We first define the following invariant condition.

Definition 5 (Invariant Condition). *The probability that the value $\widetilde{\sigma}'$ committed in com_2 by $\mathcal{A}_{\mathsf{mim}}$ is equal to $\widetilde{\sigma}$ committed in com_1 by the honest verifier is negligible.*

Note that if the invariant condition holds and $\mathcal{A}_{\mathsf{mim}}$ gives a convincing proof, we can extract the witness $\widetilde{w}$ for $\widetilde{x}$ by running the sWIAoK extractor.

At a high level, our proof goes in the following way. We start with the man-in-the-middle setting, where an honest prover $P(x, w)$ interacts with $\mathcal{A}_{\mathsf{mim}}$ in the left interaction, and $\mathcal{A}_{\mathsf{mim}}$ proves to an honest verifier V for a statement $\widetilde{x} \neq x$ in the right. We will build a sequence of hybrids, where we gradually substitute $P(x, w)$ and $V(\widetilde{x})$ with our simulator. Between each pair of adjacent hybrids, we show that the view of $\mathcal{A}_{\mathsf{mim}}$ does not change *and* that the invariant condition holds. In the last hybrid, we do not need the real witness w in the left interaction, and we can extract $\mathcal{A}_{\mathsf{mim}}$'s witness $\widetilde{w}$ via the sWIAoK extractor (we are guaranteed to extract $\widetilde{w}$ because of the invariant condition). With the extracted $\widetilde{w}$, our simulator can give a "straight-line" proof for the statement $\widetilde{x}$ to V, which completes the proof of non-malleability. Next, we describe the hybrids.

Hybrid H_0. This is the real execution of the MIM game. Specifically, H_0 sets up the left and right executions for $\mathcal{A}_{\mathsf{mim}}$ with $P(x, w)$ and V, respectively. H_0 outputs the joint view of $\mathcal{A}_{\mathsf{mim}}$ containing both left and right executions.

Invariant condition. If the invariant condition does not hold, then consider the prover machine P^* which behaves identically to H_0 except that it forwards the right ExtCom to an external committer. Using this P^* we can violate the hiding of ExtCom by extracting the value committed in the right ENMC.

Hybrid H_1. This hybrid is identical to H_0, except that whenever the left ExtCom is accepting, H_1 extracts the committed value σ in the left ExtCom. If the extractor fails ($\sigma = \perp$), H_1 outputs $\perp$ and halts; otherwise it continues as H_0.

$H_0 \overset{\text{s}}{\approx} H_1$. The outputs of H_0 and H_1 differ only when $\sigma = \perp$; and due to the extractability of ExtCom, that happens with only negligible probability.

Invariant condition. The invariant condition holds in H_1 since it holds in H_0 and the two hybrids are statistically close.

Remark 1. Note that the above proofs for both indistinguishability and invariant condition are independent of $\mathcal{A}_{\mathsf{mim}}$'s scheduling of the messages. Thus, they also hold in the non-synchronous scenario.

Hybrid H_2. This hybrid is identical to H_1, except that H_2 sets $\sigma' = \sigma$ in **Stage-2** ENMC on left.

$H_1 \overset{\text{c}}{\approx} H_2$ follows immediately from the computational-hiding property of ENMC.

Invariant condition. The fact that the invariant condition holds can be reduced to the *non-malleability* of ENMC. Specifically, we consider a man-in-the-middle adversary $\mathcal{A}_{\mathsf{ENMC}}$ for ENMC that acts as follows: $\mathcal{A}_{\mathsf{ENMC}}$ internally runs H_2 except that it obtains the left ENMC execution from an outsider committer on the left and forwards the right ENMC interaction to an external receiver. Furthermore, the external committer commits as follows: recall that H_2 already has the

extracted value σ before the left ENMC begins; $\mathcal{A}_{\mathsf{ENMC}}$ forwards $\sigma'_0 = 0^\lambda$ and $\sigma'_1 = \sigma$ to the external committer who then commits to one of them at random. $\mathcal{A}_{\mathsf{ENMC}}$ halts when H_2 halts. Now consider a distinguisher D (that incorporates the above adversary $\mathcal{A}_{\mathsf{ENMC}}$), and by definition of non-malleability, receives the value $\mathcal{A}_{\mathsf{ENMC}}$ commits to in the right interaction, say $\widetilde{\sigma}$. Clearly, if the invariant condition does not hold in H_2 then the distribution of $\widetilde{\sigma}$ is different depending on whether $\mathcal{A}_{\mathsf{ENMC}}$ receives commitment to σ'_0 or σ'_1. This condition can be tested by D (which incorporates $\mathcal{A}_{\mathsf{ENMC}}$), thus violating the non-malleability of ENMC.

Remark 2. Observe that in the non-synchronous case, the proof of indistinguishability will go through, but the proof of invariant condition will not. This is because the extraction of α on left from ExtCom may rewind some parts of ENMC on right, and this is not allowed by the non-malleability definition. We will deal with this issue in Sec. 4.2.

Hybrid H_3. Identical to H_2 except that it switches from real witness w to the trapdoor witness (i.e., values and randomness corresponding to $\sigma' = \sigma$) in the **Stage-4** sWIAoK on left.

$H_3 \overset{\text{s}}{\approx} H_2$ follows directly from the statistical WI property of sWIAoK.

Invariant condition. Since we are in the synchronous setting, the invariant condition holds since the executions in the two hybrids are identical up to the end of **Stage-2**, at which point the invariant condition is already determined; any changes after that stage have no effect on the invariant condition.

Remark 3. As in Rmk. 2, in the non-synchronous case, the argument for indistinguishability still holds, but the argument for the invariant condition will require extra caution. This is because the left sWIAoK may get aligned with the right ENMC so that the switch of witness may affect the invariant condition. We will deal with this issue in Sec. 4.2.

Simulator for Non-Malleability. The indistinguishability among the above hybrids implies that: if $\mathcal{A}_{\mathsf{mim}}$ gives a convincing proof in the right interaction of H_0, it should also give a convincing proof in the right interaction of H_3. We construct a simulator Sim in the following way. Given a man-in-the-middle adversary $\mathcal{A}_{\mathsf{mim}}$, Sim first invokes H_3 with $\mathcal{A}_{\mathsf{mim}}$. If $\mathcal{A}_{\mathsf{mim}}$ indeed gives a convincing proof in the right interaction, Sim extracts $\mathcal{A}_{\mathsf{mim}}$'s witness $\widetilde{w}$ from sWIAoK on the right execution; otherwise, Sim aborts. The invariant condition in H_3 guarantees that Sim can extract such a $\widetilde{w}$. With $\widetilde{w}$, Sim then executes protocol $\langle P, V\rangle_{\mathsf{NMZK}}$ (in "straight-line") with an honest verifier. It convinces the honest verifier with roughly the same probability as $\mathcal{A}_{\mathsf{mim}}$ (except for negligible difference due to Sim's failure in extracting $\widetilde{w}$). This finishes the proof of non-malleability *against synchronous adversaries.*

4.2 Non-Malleability against Non-Synchronous Adversaries

As mentioned in Rmk. 1 to 3, the proofs for indistinguishability among *all* hybrids, as well as the invariant condition for H_0 and H_1, remain unchanged in the non-synchronous setting. Therefore, we only need to prove the invariant conditions for H_2 and H_3, which will be done in the sequel. (We first show the proof for H_3 since it is simpler.)

The Invariant Condition for H_3. Recall that the witness indistinguishability of the sWIAoK is *statistical*. It follows that the invariant condition must hold in H_3 for non-synchronous adversaries as well. If not, an exponential time distinguisher can recover the value committed by $\mathcal{A}_{\mathsf{mim}}$, thus breaks the statistical WI by testing whether the invariant condition.

The Invariant Condition for H_2. Before giving the formal lemma and proof, we provide the high-level idea. As mentioned in Rmk. 2, the problem happens if the $\mathcal{A}_{\mathsf{mim}}$ interleaves the left ExtCom messages with the right ENMC messages. In such a schedule, we cannot reduce the invariant condition to the non-malleability of ENMC without rewinding the outside challenger in ENMC's man-in-the-middle game. Recall that both H_1 and H_2 rewind the left ExtCom to extract the committed value σ.

We first note that if the reduction can simulate the receiver-to-committer messages in ENMC, then there is no issue during rewinding since in the right interaction, the reduction can forward messages between $\mathcal{A}_{\mathsf{mim}}$ and the outside challenger to the "main thread" and simply simulate them in "rewinding" threads. This (informally-explained) property is indeed satisfied by our $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ commitment (Prot. 1).

In the following, we show the formal claim and its proof.

Claim 1. *The invariant condition holds in Hybrid H_2 described in Sec. 4.1 for non-synchronous adversaries.*

Proof. This proof relies on the special structure of ENMC (when instantiated as the $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ protocol shown in Prot. 1). We will refer to different rounds of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$, which are recalled below for convenience (see and compare with Prot. 1):

- **(1):** R sends the first message for Naor's commitment, which consists of public coins only.
- **(2):** C sends the second message of Naor's commitment.
- **(3):** R sends some (public) random vectors as his challenge.
- **(4):** C responds to R's challenges. C also sends the first message (which consists of some public coins that specifies a CRHF) of a Ligero' instance, which is used for consistency proof.
- **(5)-(10):** These are rounds 2 to 7 of Ligero' between C and R. Note that **(5)** is the (statistically-hiding) commitments to verifier's random challenge Γ_1 and Γ_2; **(7)** is R's decommitment to Γ_1 and **(9)** is R's decommitment to Γ_2.

With the structure of ENMC in mind, we now start to prove Claim 1. First, observe that ExtCom has only one "slot" that is rewound to extract σ. Therefore, we only need to worry about the schedule where some messages of the right ENMC are "nested" in this slot. In the following, we show that the invariant condition hold for all schedules.

In the following, we use **(***i***)** ($i \in [10]$) to denote the i-th step of the right ENMC (as recalled above). We denote the first message of the rewindable slot in the left ExtCom as top, and the last message as bottom. See Fig. 1 for an illustration of these notations. Note that in Fig. 1, no messages can appear between "adjacent messages" of the right ENMC, for example, message **(2)**-**(3)**, **(6)**-**(7)** etc. This is because honest parties send their next message as soon as they receive the previous message.

Easy Cases. First, note that if bottom happens before **(1)**, we can rewind the slot without rewinding the right ENMC. Therefore, the same proof for the invariant condition in H_2 in the synchronous setting also applies here. Also, it is an easy case when **(10)** happens before bottom. In this case, $\mathcal{A}_{\mathsf{mim}}$ cannot generate the right ENMC messages based on the left ENMC interactions, since the left ENMC has not started yet. Therefore, the invariant condition holds automatically. Another easy case is when **(1)** gets nested in the slot. In such a case, rewinding the slot will cause a fresh execution of the right ENMC, so it will not cause any problem when we try to reduce the invariant condition to the non-malleability of ENMC. At a high level, this is because we can always forward the messages when we do the last rewinding to the outside non-malleability challenger in the reduction. But we suppress the details here since we will provide a formal argument of such type when we handle the hard cases next.

Hard Cases. We now focus on the remaining schedules (beyond those discussed in **Easy cases**). These schedules consist of the situations where

- **(1)** happens before top, *and*
- **(10)** happens after bottom.

There are 10 such cases in total. Since these 10 schedules can be handled via similar arguments, in the following, we will use the one in Fig. 1 as a representative to present a full proof, and then discuss how to extend the same proof to the remaining 9 cases in the full version [53].

For the schedule shown in Fig. 1, we build a man-in-the-middle adversary $\mathcal{A}_{\mathsf{ENMC}}$ attacking the non-malleability of ENMC. Recall that in the non-malleability game, the man-in-the-middle adversary $\mathcal{A}_{\mathsf{ENMC}}$ talks to an honest committer in the left, and to an honest receiver in the right. We will refer to them as the left challenger and right challenger respectively. Our $\mathcal{A}_{\mathsf{ENMC}}$ acts in the following way:

1. $\mathcal{A}_{\mathsf{ENMC}}$ starts by running the hybrid experiment H_2 *internally* with $\mathcal{A}_{\mathsf{mim}}$ up to the step right before **(1)**. It then invokes the right challenger for the non-malleability game of ENMC, and forwards the messages between the

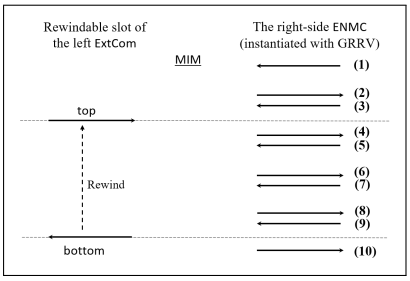

Fig. 1: Special Schedules in the Non-Synchronous Scenario

challenger and $\mathcal{A}_{\mathsf{mim}}$ as the right interaction. It plays the left interaction in the same way as the simulator in H_2, until the execution reaches top for the first time.

2. $\mathcal{A}_{\mathsf{ENMC}}$ now needs to execute the slot $(\mathsf{top}, \mathsf{bottom})$ in the "main-thread", and then rewind this slot for (w.l.o.g.) $k = \mathsf{poly}(\lambda)$ times to extract the σ value in the left interaction. To do that, $\mathcal{A}_{\mathsf{ENMC}}$ proceeds as follows:

   (a) For the main-thread execution, $\mathcal{A}_{\mathsf{ENMC}}$ plays the right interaction by forwarding messages between $\mathcal{A}_{\mathsf{mim}}$ and the outside right challenger.

   (b) From the 1st to the k-th rewinding, $\mathcal{A}_{\mathsf{ENMC}}$ will prepare the right ENMC incoming messages (i.e. **(5)**, **(7)**, and **(9)**) *by himself*, instead of forwarding them between $\mathcal{A}_{\mathsf{mim}}$ and the outside right challenger. To do that, $\mathcal{A}_{\mathsf{ENMC}}$ samples *fresh* Γ_1 and Γ_2, and commits to them as message **(5)**; it sends the honest decommitments to (the fresh) Γ_1 as message **(7)**; similarly, it sends the honest decommitments to (the fresh) Γ_2 as message **(9)**. We emphasize that $\mathcal{A}_{\mathsf{ENMC}}$ can indeed decommit to them because the commitments in **(5)** (in these rewinding threads) are generated by himself.

   Note that the simulated messages during rewinding have identical distribution as the main-thread **(5)**, **(7)**, and **(9)**, which guarantees that $\mathcal{A}_{\mathsf{mim}}$'s view does not change. Thus, after the above rewindings, σ can be extracted except for negligible probability, for which $\mathcal{A}_{\mathsf{ENMC}}$ just halts outputting $\perp$.

3. $\mathcal{A}_{\mathsf{ENMC}}$ continues the internal (main-thread) interaction until the left ENMC starts. He then invokes the outside left challenger by sending the values $\sigma'_0 = 0^\lambda$ and $\sigma'_1 = \sigma$. Then, ENMC forwards the messages between $\mathcal{A}_{\mathsf{mim}}$ and the outside left challenger and $\mathcal{A}_{\mathsf{mim}}$ as the left ENMC interaction. In the right interaction, ENMC acts as the simulator in H_2 except that when $\mathcal{A}_{\mathsf{mim}}$ sends the message **(10)**, it forwards the message to the outside right challenger.

4. $\mathcal{A}_{\mathsf{ENMC}}$ continues to finish the internal interaction with $\mathcal{A}_{\mathsf{mim}}$ as in H_2 for the remaining parts of the protocol.

Now consider a distinguisher D (that incorporates the above adversary $\mathcal{A}_{\mathsf{ENMC}}$), and by definition of non-malleability, receives the value $\mathcal{A}_{\mathsf{ENMC}}$ commits to in the right interaction, say $\widetilde{\sigma}$. Clearly, if the invariant condition does not hold in H_2 then the distribution of $\widetilde{\sigma}$ is different depending on whether $\mathcal{A}_{\mathsf{ENMC}}$ receives commitment to σ_0' or σ_1'. This condition can be easily tested by D (since it incorporates $\mathcal{A}_{\mathsf{ENMC}}$), thus violating the non-malleability of ENMC.

The above argument proves Claim 1 for the special scheduling shown in Fig. 1. Due to space constraints, we postpone the discussion for the other 9 schedules to the full version [53]. □

4.3 Generalization to "Almost Public-Coin" Statistically ZK

In this part, we take another look at the proof in Sec. 4.2 with the following purpose: in Sec. 4.2, we proved the invariant condition in H_2, relying on the special structure of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$. In particular, we assumed that the **Consistency Proof** stage of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ is conducted by Ligero'. However, we argue that Ligero' can be replaced by any "almost public-coin" (explained below) statistically zero-knowledge argument.

Motivation. Before delving into the details, let us first explain why we want to generalize the proof to almost public-coin ZK protocols: While Ligero' is efficient, using it directly in the **Consistency Proof** stage of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ results in unacceptable running time. This is because the language $L^{\rho}_{\mathsf{consis}}$ (defined toward the end of Prot. 1) has a huge circuit size. As mentioned in Sec. 1.2, we will (in Sec. 5.3) introduce the new idea of converting $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ to an *instance-based* non-malleable commitment to achieve better efficiency. Looking ahead, the instance-based $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ shares the same structure of the original $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$, with the only difference being that the **Consistency Proof** stage is not conducted by Ligero' anymore. Instead, it will be done using a customized statistical ZK protocol called Π'_{OR}, which we construct by applying (a modified version of) the OR-composition technique [22] on Ligero (i.e., the honest-verifier version of Ligero'). We need to show that the same proof in Sec. 4.2 will still go through when we replace (the original) $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ with this instance-based $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ (i.e., when we replace Ligero' in the **Consistency Proof** stage with Π'_{OR}). Fortunately, this is possible because Π'_{OR} shares the same structure as Ligero', in terms of the application in Sec. 4.2. In particular, Π'_{OR} also enjoys the same "almost public-coin" property of Ligero', and this is exactly why the same proof in Sec. 4.2 can be applied when we replace Ligero' with Π'_{OR}. The purpose of this subsection is to distill this "almost public-coin" property and explain how it helps in the proof in Sec. 4.2.

Almost Public-Coin Protocols. Let us summarize how the proof in Sec. 4.2 makes use of the structure of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$. As we mentioned in the beginning of Sec. 4.2, $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ has 10 rounds that can be understood as two stages:

1. **Commit Stage:** This includes rounds **(1)** to **(4)**, and

2. **Consistency Proof:** This includes rounds **(4)** to **(10)**, which is exactly the statistically ZK protocol Ligero'.

We emphasize that all the receiver's messages are public coins except for rounds **(5)** and **(7)**, which constitute the commitment and corresponding decommitment to some random coins. This public-coin property is the main reason that $\mathcal{A}_{\mathsf{ENMC}}$ works properly: in Step 2, $\mathcal{A}_{\mathsf{ENMC}}$ needs to simulate the receiver's message in rewinding threads; because all the receiver's messages (except for rounds **(5)** and **(7)**) are public-coin, $\mathcal{A}_{\mathsf{ENMC}}$ can simply sample them freshly for each rewinding; moreover, round **(5)** (resp. **(7)**) is a commitment (resp. the corresponding decommitment) to random coins, so $\mathcal{A}_{\mathsf{ENMC}}$ can also sample and commit to (resp. decommit honestly to) random coins itself. Therefore, the rewinding threads can be shown to be identically distributed as the main thread.

In light of the above, it is clear that the $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ can be replaced with any ENMC that enjoys the above public-coin property. In particular, the **Commit Stage** of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ is public-coin by design; the **Consistency Proof** stage, when implemented with Ligero', is public-coin (again, except for **(5)** and **(7)** as discussed above) because Ligero' is obtained in a special way: it is obtained by applying the Goldreich-Kahan transform on the honest-verifier version Ligero, which is a public-coin protocol.

Looking ahead, our Π'_{OR} enjoys the above public-coin property. As we will show in Sec. 5.3, Π'_{OR} is obtained by applying Goldreich-Kahan transform on a protocol Π_{OR} (which will appear in Sec. 5.2), which is also a public-coin honest-verifier ZK argument. Therefore, the above argument applies.

In summary, when we replace $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ with its instance-based version, the same proof in Sec. 4.2 will still go through.

5 Improving Efficiency through Fake Executions

5.1 Road Map of This Section

In this section, we describe how to instantiate our NMZK protocol $\langle P, V\rangle_{\mathsf{NMZK}}$ (shown in Prot. 2) to achieve concrete efficiency. The major bottlenecks are:

1. Step 4 of $\langle P, V\rangle_{\mathsf{NMZK}}$ is a statistical WIAoK on the OR-composition of the statement x and a trapdoor statement (let us denote it as $(x \vee x_{\mathsf{tr}})$). This proof is non-black-box on the Step 2 commitments and involves expensive $\mathcal{NP}$ reduction.
2. Step 2 of $\langle P, V\rangle_{\mathsf{NMZK}}$ is instantiated with $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ (Prot. 1), whose **Consistency Proof** step involves an expensive ZK proof.

To address Item 1, we want to employ the OR-composition technique in [22] to construct the desired sWIAoK from Ligero. This will allow the prover to finish the proof for $(x \vee x_{\mathsf{tr}})$ by conducting a (light) proof for x, and running the fast Ligero simulator HVSim for the x_{tr} part. This will be much more efficient than running Ligero on $(x \vee x_{\mathsf{tr}})$ directly. However, this approach encounters obstacles:

Ligero does not have the properties required by [22]. We show how to solve related problems in Sec. 5.2.

To address Item 2, we wish to reuse the OR-composition technique described above. But it does not immediately apply because the target statement of the **Consistency Proof** does not have the $(x \vee x_{\mathsf{tr}})$ structure; instead, it is a single statement $x_{\mathsf{com}} \in L^{\rho}_{\mathsf{consis}}$, which is related to some vector of commitments[6]. Running Ligero for x_{com} is prohibitively expensive. To handle this issue, observe that this $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ protocol is executed as a part of our $\langle P, V\rangle_{\mathsf{NMZK}}$ protocol on some statement x_{zk}. Therefore, we change the statement of **Consistency Proof** to $(x_{\mathsf{zk}} \vee x_{\mathsf{com}})$, and then use the above OR-composition technique to boost the efficiency. We denote this extended non-malleable commitments as *instance-based non-malleable commitments* (IB-NMC). We elaborate on the above idea in Sec. 5.3.

Non-Malleability from Simulation-Soundness. Unfortunately, the above strategy induces an extra problem—replacing the Step 2 ENMC by the above instance-based version (i.e. the IB-NMC) jeopardizes the security of $\langle P, V\rangle_{\mathsf{NMZK}}$ (Prot. 2). Specifically, it is not clear whether the resulting protocol is still non-malleable. However, we will be able to prove that it is a *simulation-sound* ZK protocol (which is already sufficient for many applications). Finally, we show in Sec. 5.4 (resp. Sec. 5.5) how to use this simulation-sound ZK protocol to obtain non-malleable ZK protocols (resp. non-malleable commitments), with (almost) no efficiency overhead.

5.2 OR-Composition of Ligero

The OR-composition [22] was originally designed for Σ protocols, i.e., 3-round public-coin HVZK protocols with *special soundness*, which requires that a witness can be extracted from two convincing transcripts with distinct challenges. To prove an OR statement $x \vee x'$, the OR-composition invokes a parallel execution of two Σ-protocol instances: (a_1, b_1, c_1) for proving x and (a_2, b_2, c_2) for proving x', which are called the left and right execution respectively. But the verifier sends only a single round-2 challenge b; the prover has the freedom to "decompose" it as $b = b_1 \oplus b_2$ to finish the two parallel executions. The prover may only have a witness for, say, the x part; since it can always "equivocate" one share of b, it will first "finish" (in other words, fake) the left execution by running the HVZK simulator for the Σ-protocol by setting b_2 in advance; it can answer any $b_1 = b \oplus b_2$ as it has the witness for x.

We want to apply the above OR-composition to Ligero. However, Ligero is not a Σ-protocol—it has six rounds (i.e., two challenge-response slots). Indeed, it is known that straightforward generalization of OR-composition to multi-slots protocols (i.e., the original OR-composition is applied on each slot separately) will yield an *unsound* protocol.

[6] Recall that the language $L^{\rho}_{\mathsf{consis}}$ is defined toward the end of $\Pi^{\mathsf{Mini}}_{\mathrm{BGRRV}}$ (Prot. 1).

The First Attempt. In more detail, recall that Ligero's messages are denoted as $(h, a, b, c, \widetilde{b}, \widetilde{c})$, where $(h, b, \widetilde{b})$ are nothing but public random coins. If we do the straightforward generalization of the above OR-composition (to prove an OR statement $x \vee x'$), it will work as follows: assuming P knows witness w for x, P uses $\mathsf{HVSim}(x')$ to simulate a proof $(h_2, a_2, b_2, c_2, \widetilde{b}_2, \widetilde{c}_2)$ for the x' part (because P does not have witness for it). Meanwhile, P generates the proof for x honestly, in the following manner: V sends h and P derives h_1 as $h_1 = h \oplus h_2$; P runs the honest Ligero prover's algorithm on input (x, w) to generate a_1, assuming the first Ligero verifier's message is h_1. Similarly, when V sends b (resp. $\widetilde{b}$), P will set $b_1 = b \oplus b_2$ (resp. $\widetilde{b}_1 = \widetilde{b} \oplus \widetilde{b}_2$), and compute the response c_1 (resp. $\widetilde{c}_1$) using the honest Ligero prover's algorithm (as it has witness w for x).

However, the above approach suffers from the following "cross attack": Since P^* has the opportunity to decide how to decompose h, b, and $\widetilde{b}$, it can pick a bad b_1 and a bad $\widetilde{b}_2$. That is, a cheating prover can choose malicious challenges in the first slot of the *left* execution and the second slot of the *right* execution, and there is no soundness guarantee for Ligero when a malicious prover can control (even) one challenge out of the two slots.

Solution. To resolve this problem, we ask P to commit to its decomposition in advance. More accurately, we ask P to generate $\mathsf{com} = \mathsf{SHCom}(h_2 \| b_2 \| \widetilde{b}_2; r)$ at the very beginning of the protocol, where SHCom is a statistically-hiding commitment. Then, we continue as the above. At the end of the execution, we ask P to give a statistical WI argument of knowledge sWIAoK for the following statement:

- com is committing to either $(h_1, b_1, \widetilde{b}_1)$ or $(h_2, b_2, \widetilde{b}_2)$.[7]

Intuitively, due to the (knowledge) soundness of sWIAoK, P^* cannot conduct the above "cross attack" anymore.

We denote this protocol as Π_{OR}. Due to space constraints, we put the formal description of Π_{OR} in the full version [53], where we also provide the complete security proof. Here, we want to emphasize that this approach invokes very small efficiency overhead compared with the plain OR-composition described in **The First Attempt**: what we add is simply a statistically-hiding commitment and a sWIAoK for its consistency. Using a modified version of Ligero as the underlying sWIAoK (see [53, Appendix C.3]), this only adds an extra computation cost of 32 milliseconds and an extra communication cost of 6.4MB. See [53, Appendix C.2] for more details.

Regarding Malicious-Verifiers ZK. It is not hard to see that the above Π_{OR} is also an honest-verifier ZK argument (of knowledge). Using the Goldreich-Kahan technique [38] (as done in [46, 2]), we can convert it to a fully-secure ZK argument, i.e., against malicious verifiers. We denote the resulting protocol as Π'_{OR}, and present the full description of it in [53, Protocol 11]. Looking ahead,

[7] Note that $(h_1, b_1, \widetilde{b}_1)$ and $(h_2, b_2, \widetilde{b}_2)$ will be known to V when the protocol reaches the final sWIAoK stage.

Π'_{OR} will be used in the instance-based non-malleable commitment in the next subsection (in Prot. 3).

5.3 Instance-Based Non-Malleability

Recall that we use $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ (Prot. 1) as our ENMC. The primary efficiency bottleneck in $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ is the consistency proof, which is done using Ligero'. Since an honest committer is never cheating, our goal is to provide the prover an easier way to get through this proof. Toward this goal, we first show an *instance-based* version of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$, denoted as $\langle C_L, R_L \rangle$. The instance-based version simply gives the option of using a witness for a true statement in the consistency proof phase of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$. At a high level, the parties get a statement x as input which may or may not be true. If x is true, the committer can additionally take as input a witness $w \in \mathcal{R}_L(x)$ and succeed in the proof phase by using w instead of completing the consistency proof for any message m. This allows the honest prover to fake the ENMC execution using a faster simulator thanks to the OR-composition. If x is false, the committer commits to a valid value m. It is also possible to do both: commit to m properly and execute consistency proof as well as proof for x. We present the full construction in Prot. 3,[8] and establish its security in Lem. 2 and 3.

Protocol 3: $\langle C_L, R_L \rangle(x)$: Instance-Based Non-Malleable Commitment

Instance-based $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ is the following commitment scheme, denoted as $\langle C_L, R_L \rangle$, defined for an arbitrary $\mathcal{NP}$ language L: the common input to both algorithms is a statement x; in addition, C_L takes a (private) auxiliary input that is either of the form $(m, \perp)$ or $(\perp, w)$ where w is a witness for $x \in L$. Recall that $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ is denoted by $\langle C, R \rangle$ and depicted in Prot. 1. The protocol proceeds in two phases:

- **Commit Stage:** In this stage R_L proceeds identically to algorithm R of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ and let ρ be its first message. For input $(m, \perp)$, C_L proceeds exactly as C proceeds in the commit stage on input m. For input $(\perp, w)$, C_L simply sends *random* values of appropriate size as the second and fourth messages of the commit stage (when interacting with R_L). Recall that the execution of the **Commit Stage** of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ will yield messages $\widehat{\mathbf{m}}$, $\widehat{\mathbf{r}}$, and $\{w_i\}_{i\in[n]}$) (see Prot. 1). We denote $\mathsf{st} \coloneqq (\widehat{\mathbf{m}}, \widehat{\mathbf{r}}, \{w_i\}_{i\in[n]})$.
- **Proof Stage:** In this stage, C_L proves that $(x, \mathsf{st}) \in L \vee L^{\rho}_{\mathsf{consis}}$ using Π'_{OR}, i.e., the fully ZK version of Π_{OR} (see [53, Protocol 11]). For input $(m, \perp)$, C_L uses the simulator HVSim for the left part (i.e., for x), and completes right part (i.e., for st) honestly by using the witness for st (from the first phase). For input $(\perp, w)$ it uses w to succeed in the left part of the proof and simulator HVSim to succeed in the right part.

If the common statement is fixed to x, we denote the instance-based $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ by $\langle C_L, R_L \rangle(x)$. The executions corresponding to inputs $(m, \perp)$ will be called **real or honest executions**, and those corresponding to $(\perp, w)$, **fake or simulated executions** of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ (or ENMC).

Lemma 2. *Let L be an $\mathcal{NP}$ language. For every $x \notin L$ protocol $\langle C_L, R_L \rangle(x)$ (Prot. 3) is an extractable non-malleable commitment scheme.*

Proof. We observe that for every $x \notin L$, the **Proof Stage** of the protocol is a ZK argument for $\mathsf{st} \in L^{\rho}_{\mathsf{consis}}$ (i.e., consistent execution of the commit stage). In

[8] We warn that this version cannot be used in our NMZK protocol yet. See Sec. 5.4.

this case, $\langle C_L, R_L \rangle(x)$ is simply an instantiation of the original $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ protocol. The claim then follows from the security of $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$. □

Lemma 3. *Let L be an $\mathcal{NP}$ language with witness relation $\mathcal{R}_L$. For every message m and every $(x, w) \in \mathcal{R}_L$, the following holds:*

$$\{\mathsf{view}_0 \leftarrow \langle C_L((m, \perp)), R_L \rangle(x) : \mathsf{view}_0\} \overset{c}{\approx} \{\mathsf{view}_1 \leftarrow \langle C_L((\perp, w)), R_L \rangle(x) : \mathsf{view}_1\}.$$

Proof. This lemma follows from the following two observations: (i) committer's messages in the commit stage are *pseudorandom* (since second message of Naor's commitment is pseudorandom), and (ii) the proof stage is WI (it is indeed ZK). Since the proof follows from a standard hybrid argument, we omit the details. □

Remark 4 (On Efficiency). It is worth noting that if x admits a fast Ligero proof, then fake executions are faster than the real executions since the simulator for Ligero for the right part (i.e., the real consistency proof for st) is much faster than the prover. As mentioned in Sec. 1.2, this is how we manage to obtain significant improvement on the efficiency.

5.4 Efficient Simulation-Sound Zero-Knowledge

The main benefit of the instance-based $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ in Prot. 3 is that if $x \in L$ admits fast proofs, it can be used in place of standard $\Pi^{\mathsf{Mini}}_{\text{BGRRV}}$ in our NMZK protocol. Unfortunately, the resulting protocol is not a NMZK for true x! Nevertheless, the resulting protocol is *simulation-sound* (as per Def. 3), and equally importantly, *efficient*. We refer to this protocol by Π_{SS} and specify it in Prot. 4.

Protocol 4: Π_{SS}: Simulation-Sound ZKAoK

The common input is x and prover's input is a witness w for $x \in L$, where L is the desired $\mathcal{NP}$ language. This protocol is identical to protocol $\langle P, V \rangle_{\mathsf{NMZK}}$ (Prot. 2) except that the Step 2 ENMC is replaced with the instance-based non-malleable commitment (Prot. 3) with the following inputs: the common input is x and committer's auxiliary input in the **Proof Stage** of the commitment is $(\perp, w)$. Observe that the honest prover only performs a simulated execution of the non-malleable commitment.

Theorem 2. *Protocol Π_{SS} (Prot. 4) is a simulation-sound zero-knowledge argument of knowledge.*

Due to space constraints, we postpone the proof of Thm. 2 to the full version [53].

5.5 Putting It All Together: Fast NMZK and NMCom

Now we show how to get efficient and full-fledged non-malleable zero-knowledge and commitment protocols with the help of our efficient simulation-sound ZKAoK protocol Π_{SS} and the statistically WIAoK protocol Π_{OR}.

Fast NMZK Protocol. We present our final NMZK protocol in Prot. 5. At a high level, the prover in Prot. 5 sets up a "trapdoor statement" in the form of a commitment cm, and proves using Π_{SS} that cm is a commitment to 0. Later, the prover proves using protocol Π_{OR} that either the statement is true or that cm is a commitment to 1. The honest prover always commits to 0 and thus remains fast. The simulator commits to 1 instead. The security of Prot. 5 can be proven Following a similar proof as that of Lem. 1. Due to space constraints, we postpone the proof to the full version [53].

Protocol 5: $\langle P, V\rangle_{\mathsf{final}}$: Non-Malleable ZKAoK

The common inputs are statement x, tag id, and security parameter λ. Prover's private input is a witness $w \in R_L(x)$, where L is the desired $\mathcal{NP}$ language. The protocol proceeds as follows:

1. P commits to 0^λ using 2-round Naor commitment; let ρ be the first message of this commitment and $\mathsf{cm} = \mathsf{Com}_\rho(0^\lambda)$ the second message.
2. P and V execute Π_{SS} with tag id, where P proves that cm is a valid commitment to 0^λ.
3. P and V execute Π_{OR}, where P proves that:
   - $x \in L$, **or**
   - cm is a valid commitment to 1^λ, i.e., $(\mathsf{cm}, 1^\lambda) \in L_{\mathsf{Com}_\rho}$.

Fast NMCom Protocol. Our non-malleable commitment protocol is presented in Prot. 6. At a high level, Prot. 6 works in the same way as the non-malleable zero-knowledge protocol above, except that x is replaced with a commitment to the desired value. Its security proof follows closely from the proof Lem. 1. The details are omitted.

Protocol 6: $\langle C, R\rangle_{\mathsf{final}}$: Non-Malleable Commitment

The common input is a tag id and the security parameter λ. Private input of the committer is a value $v \in \{0,1\}^\lambda$. The protocol proceeds as follows:

1. C commits to v using two-round Naor commitment; let R's first message be ρ, and $c = \mathsf{Com}_\rho(v)$ denote the second message.
2. C further commits to 0^λ using ρ as first message. Let $\mathsf{cm} = \mathsf{Com}_\rho(0^\lambda)$.
3. C proves that cm is valid commitment to 0^λ using Π_{SS} with tag id.
4. C proves using Π_{OR} that:
   - there exists v such that c is a valid commitment to v, i.e., $(c, v) \in L_{\mathsf{Com}_\rho}$, **or**
   - cm is a valid commitment to 1^λ, i.e., $(\mathsf{cm}, 1^\lambda) \in L_{\mathsf{Com}_\rho}$.

References

1. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. In: Shmoys, D.B. (ed.) 46th ACM STOC. pp. 774–783. ACM Press (May / Jun 2014). https://doi.org/10.1145/2591796.2591804
2. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 2087–2104. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3134104

3. Badrinarayanan, S., Goyal, V., Jain, A., Kalai, Y.T., Khurana, D., Sahai, A.: Promise zero knowledge and its applications to round optimal MPC. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 459–487. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_16
4. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS. pp. 106–115. IEEE Computer Society Press (Oct 2001). https://doi.org/10.1109/SFCS.2001.959885
5. Barak, B.: Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In: 43rd FOCS. pp. 345–355. IEEE Computer Society Press (Nov 2002). https://doi.org/10.1109/SFCS.2002.1181957
6. Barak, B., Goldreich, O.: Universal arguments and their applications. SIAM Journal on Computing **38**(5), 1661–1694 (2008)
7. Barak, B., Prabhakaran, M., Sahai, A.: Concurrent non-malleable zero knowledge. In: 47th FOCS. pp. 345–354. IEEE Computer Society Press (Oct 2006). https://doi.org/10.1109/FOCS.2006.21
8. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993). https://doi.org/10.1145/168588.168596
9. Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E.: On the concrete efficiency of probabilistically-checkable proofs. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 585–594. ACM Press (Jun 2013). https://doi.org/10.1145/2488608.2488681
10. Boldyreva, A., Cash, D., Fischlin, M., Warinschi, B.: Foundations of non-malleable hash and one-way functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 524–541. Springer, Heidelberg (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_31
11. Brenner, H., Goyal, V., Richelson, S., Rosen, A., Vald, M.: Fast non-malleable commitments. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015. pp. 1048–1057. ACM Press (Oct 2015). https://doi.org/10.1145/2810103.2813721
12. Broadnax, B., Döttling, N., Hartung, G., Müller-Quade, J., Nagel, M.: Concurrently composable security with shielded super-polynomial simulators. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 351–381. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56620-7_13
13. Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of Cryptology **13**(1), 143–202 (Jan 2000). https://doi.org/10.1007/s001459910006
14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS. pp. 136–145. IEEE Computer Society Press (Oct 2001). https://doi.org/10.1109/SFCS.2001.959888
15. Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 61–85. Springer, Heidelberg (Feb 2007). https://doi.org/10.1007/978-3-540-70936-7_4
16. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (Aug 1997). https://doi.org/10.1007/BFb0052229
17. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC. pp. 209–218. ACM Press (May 1998). https://doi.org/10.1145/276698.276741

18. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: 51st FOCS. pp. 541–550. IEEE Computer Society Press (Oct 2010). https://doi.org/10.1109/FOCS.2010.86
19. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: 34th ACM STOC. pp. 494–503. ACM Press (May 2002). https://doi.org/10.1145/509907.509980
20. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Round optimal secure multiparty computation from minimal assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part II. LNCS, vol. 12551, pp. 291–319. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64378-2_11
21. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 127–157. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63715-0_5
22. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_19
23. Damgård, I.: On σ-protocols. `http://www.cs.au.dk/~ivan/Sigma.pdf` (2002)
24. Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: 30th ACM STOC. pp. 141–150. ACM Press (May 1998). https://doi.org/10.1145/276698.276722
25. Di Crescenzo, G., Katz, J., Ostrovsky, R., Smith, A.: Efficient and non-interactive non-malleable commitment. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 40–59. Springer, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44987-6_4
26. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 601–610. ACM Press (May / Jun 2009). https://doi.org/10.1145/1536414.1536496
27. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd ACM STOC. pp. 542–552. ACM Press (May 1991). https://doi.org/10.1145/103418.103474
28. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th FOCS. pp. 523–534. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814626
29. Dwork, C., Sahai, A.: Concurrent zero-knowledge: Reducing the need for timing constraints. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 442–457. Springer, Heidelberg (Aug 1998). https://doi.org/10.1007/BFb0055746
30. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. ICS pp. 434–452 (2010)
31. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: International Conference on Cryptology in India. pp. 60–79. Springer (2012)
32. Fleischhacker, N., Goyal, V., Jain, A.: On the existence of three round zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 3–33. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_1
33. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 448–476. Springer, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49896-5_16

34. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993651
35. Giacomelli, I., Madsen, J., Orlandi, C.: Zkboo: Faster zero-knowledge for boolean circuits. In: 25th {USENIX} Security Symposium ({USENIX} Security 16). pp. 1069–1083 (2016)
36. Goldreich, O.: Foundations of Cryptography: Basic Tools, vol. 1. Cambridge University Press, Cambridge, UK (2001)
37. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge, UK (2004)
38. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology **9**(3), 167–190 (Jun 1996)
39. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. Journal of Cryptology **7**(1), 1–32 (Dec 1994). https://doi.org/10.1007/BF00195207
40. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th FOCS. pp. 102–115. IEEE Computer Society Press (Oct 2003). https://doi.org/10.1109/SFCS.2003.1238185
41. Goyal, V.: Constant round non-malleable protocols using one way functions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 695–704. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993729
42. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: 53rd FOCS. pp. 51–60. IEEE Computer Society Press (Oct 2012). https://doi.org/10.1109/FOCS.2012.47
43. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Wichs, D., Mansour, Y. (eds.) 48th ACM STOC. pp. 1128–1141. ACM Press (Jun 2016). https://doi.org/10.1145/2897518.2897657
44. Goyal, V., Richelson, S.: Non-malleable commitments using goldreich-levin list decoding. In: 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). pp. 686–699. IEEE (2019)
45. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: 55th FOCS. pp. 41–50. IEEE Computer Society Press (Oct 2014). https://doi.org/10.1109/FOCS.2014.13
46. Ishai, Y., Mahmoody, M., Sahai, A.: On efficient zero-knowledge PCPs. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 151–168. Springer, Heidelberg (Mar 2012). https://doi.org/10.1007/978-3-642-28914-9_9
47. Ishai, Y., Weiss, M.: Probabilistically checkable proofs of proximity with zero-knowledge. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 121–145. Springer, Heidelberg (Feb 2014). https://doi.org/10.1007/978-3-642-54242-8_6
48. Jain, A., Pandey, O.: Non-malleable zero knowledge: Black-box constructions and definitional relationships. In: Abdalla, M., Prisco, R.D. (eds.) SCN 14. LNCS, vol. 8642, pp. 435–454. Springer, Heidelberg (Sep 2014). https://doi.org/10.1007/978-3-319-10879-7_25
49. Katz, J., Ostrovsky, R., Smith, A.: Round efficiency of multi-party computation with a dishonest majority. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 578–595. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_36
50. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part II. LNCS, vol. 10678, pp. 139–171. Springer, Heidelberg (Nov 2017). https://doi.org/10.1007/978-3-319-70503-3_5

51. Khurana, D., Sahai, A.: How to achieve non-malleability in one or two rounds. In: Umans, C. (ed.) 58th FOCS. pp. 564–575. IEEE Computer Society Press (Oct 2017). https://doi.org/10.1109/FOCS.2017.58
52. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: 24th ACM STOC. pp. 723–732. ACM Press (May 1992). https://doi.org/10.1145/129712.129782
53. Kim, A., Liang, X., Pandey, O.: A new approach to efficient non-malleable zero-knowledge. Cryptology ePrint Archive, Paper 2022/767 (2022), https://eprint.iacr.org/2022/767, https://eprint.iacr.org/2022/767
54. Kiyoshima, S.: Round-efficient black-box construction of composable multi-party computation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 351–368. Springer, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44381-1_20
55. Lin, H., Pass, R.: Non-malleability amplification. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 189–198. ACM Press (May / Jun 2009). https://doi.org/10.1145/1536414.1536442
56. Lin, H., Pass, R.: Concurrent non-malleable zero knowledge with adaptive inputs. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 274–292. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19571-6_17
57. Lin, H., Pass, R.: Constant-round non-malleable commitments from any one-way function. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 705–714. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993730
58. Lin, H., Pass, R., Soni, P.: Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In: Umans, C. (ed.) 58th FOCS. pp. 576–587. IEEE Computer Society Press (Oct 2017). https://doi.org/10.1109/FOCS.2017.59
59. Lin, H., Pass, R., Tseng, W.L.D., Venkitasubramaniam, M.: Concurrent non-malleable zero knowledge proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 429–446. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_23
60. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 571–588. Springer, Heidelberg (Mar 2008). https://doi.org/10.1007/978-3-540-78524-8_31
61. Micali, S.: CS proofs (extended abstracts). In: 35th FOCS. pp. 436–453. IEEE Computer Society Press (Nov 1994). https://doi.org/10.1109/SFCS.1994.365746
62. Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. In: 47th FOCS. pp. 367–378. IEEE Computer Society Press (Oct 2006). https://doi.org/10.1109/FOCS.2006.43
63. Micciancio, D., Petrank, E.: Simulatable commitments and efficient concurrent zero-knowledge. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 140–159. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_9
64. Naor, M.: Bit commitment using pseudo-randomness. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 128–136. Springer, Heidelberg (Aug 1990). https://doi.org/10.1007/0-387-34805-0_13
65. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_6

66. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (Aug 2002). https://doi.org/10.1007/3-540-45708-9_8
67. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_30
68. Ostrovsky, R., Pandey, O., Visconti, I.: Efficiency preserving transformations for concurrent non-malleable zero knowledge. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 535–552. Springer, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_32
69. Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive one-way functions and applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 57–74. Springer, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_4
70. Pass, R.: On deniability in the common reference string and random oracle model. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 316–337. Springer, Heidelberg (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_19
71. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_10
72. Pass, R.: Concurrent security and non-malleability (invited talk). In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, p. 540. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19571-6_32
73. Pass, R., Rosen, A.: Concurrent non-malleable commitments. In: 46th FOCS. pp. 563–572. IEEE Computer Society Press (Oct 2005). https://doi.org/10.1109/SFCS.2005.27
74. Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 533–542. ACM Press (May 2005). https://doi.org/10.1145/1060590.1060670
75. Pass, R., Wee, H.: Black-box constructions of two-party protocols from one-way functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (Mar 2009). https://doi.org/10.1007/978-3-642-00457-5_24
76. Prabhakaran, M., Sahai, A.: New notions of security: Achieving universal composability without trusted setup. In: Babai, L. (ed.) 36th ACM STOC. pp. 242–251. ACM Press (Jun 2004). https://doi.org/10.1145/1007352.1007394
77. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS. pp. 543–553. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814628
78. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC. pp. 475–484. ACM Press (May / Jun 2014). https://doi.org/10.1145/2591796.2591825
79. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (Aug 2007). https://doi.org/10.1007/978-3-540-74143-5_12
80. Wee, H.: Zero knowledge in the random oracle model, revisited. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 417–434. Springer, Heidelberg (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_25
81. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: 51st FOCS. pp. 531–540. IEEE Computer Society Press (Oct 2010). https://doi.org/10.1109/FOCS.2010.87