Constant Round Non-Malleable Protocols using One Way Functions

Vipul Goyal
Microsoft Research, India
Email: vipul@microsoft.com

May 25, 2011

Abstract

We provide the first constant round constructions of non-malleable commitment and zero-knowledge protocols based only on one-way functions. This improves upon several previous (incomparable) works which required either: (a) super-constant number of rounds, or, (b) non-standard or sub-exponential hardness assumptions, or, (c) non-black-box simulation and collision resistant hash functions. These constructions also allow us to obtain the first constant round multi-party computation protocol relying only on the existence of constant round oblivious transfer protocols. Our primary technique can be seen as a means of implementing the previous "two-slot simulation" idea in the area of non-malleability with only black-box simulation.

A simple modification of our commitment scheme gives a construction which makes use of the underlying one-way function in a black-box way. The modified construction satisfies the notion of what we call *non-malleability w.r.t. replacement*. Non-malleability w.r.t. replacement is a slightly weaker yet natural notion of non-malleability which we believe suffices for many application of non-malleable commitments. We show that a commitment scheme which is non-malleable only w.r.t. replacement is sufficient to obtain a (fully) black-box multi-party computation protocol. This allows us to obtain a constant round multi-party computation protocol making only a black-box use of the standard cryptographic primitives with polynomial-time hardness thus directly improving upon the recent work of Wee (FOCS'10).

1 Introduction

Ever since the work of Dolev, Dwork and Naor [DDN91], obtaining efficient constructions of non-malleable protocols with small round complexity has been an important goal. The first constant round constructions of non-malleable commitments and zero-knowledge were given in the work of Barak [Bar02] (building in turn on the techniques from [Bar01]). Since then, a number of works have investigated the round complexity of non-malleable protocols. The current state of art is represented by a number of incomparable results.

- Super-constant round protocols based on one-way functions [DDN91, LP09, Wee10].
- Constant round protocols using non-standard or subexponential hardness assumptions [PPV08, PW10].
- Constant round protocols using *non-black-box simulation* techniques [Bar02, PR05b]. Subsequent to the work of [Bar02], an improved construction was later obtained by Pass and Rosen [PR05b] assuming only collision resistant hash functions. The primary disadvantage of these constructions is that the non-black-box simulation techniques used build on expensive machinery like the PCP theorem.

To date there are no known constructions of constant round non-malleable commitments or zero-knowledge using black-box simulation under any standard polynomial time hardness assumption.

We resolve this open question in this work and provide constant round constructions for both non-malleable commitment as well zero-knowledge protocols using only one way functions (OWF). This simultaneously improves upon all of these previous works. Once we get a construction of non-malleable zero-knowledge, it is possible to obtain constant round secure multi-party computation (MPC) protocols using known techniques [BMR90]. This allows us to prove the following unconditional equivalence (previously unknown even using non-black-box simulation):

Theorem 1. *A constant round (semi-honest secure) oblivious transfer protocol is necessary and sufficient to obtain a constant round secure multi-party computation protocol (unconditionally).*

The above constant round constructions (of non-malleable commitments/zero-knowledge and MPC) make a non-black-box usage of the underlying OWF (even though the proof of security is black-box w.r.t. the code of the adversary). An important step towards obtaining practical protocols is to obtain constructions making a *black-box use* of the underlying cryptographic primitives. Towards that end, we note that while our basic construction of non-malleable commitments uses the underlying OWF in a non-black-box way, a slight variant makes only a black-box usage of the OWF. The variant so obtained satisfies a slightly weaker notion of non-malleability which still turns out to be sufficient for many applications including to construct an MPC protocol. This allows us to obtain the first constant round MPC protocol making only a black-box use of standard cryptographic primitives with polynomial-time hardness.[1] This improves upon the work of Wee[Wee10] and Ishai et. al. [IKLP06].[2]

Theorem 2. (Informal statement) *There exists a (fully) black-box construction of a constant round MPC protocol from a variety of standard cryptographic primitives with polynomial-time hardness (such as lossy encryption schemes, homomorphic encryption schemes, dense cryptosystems or certifiable enhanced trapdoor permutations).*

1.1 Our Techniques

Our primary technique can be seen as a way of "de-non-black-box-izing" the two slot simulation technique of Pass [Pas04] (which in turn was used in [PR05b] to construct constant round non-malleable protocols). That is, we obtain a protocol with properties similar to that of Pass while relying only on OWF and making only a black-box use of the adversary machine in the proof of security. Then similar to [PR05b], we obtain constant round constructions of non-malleable commitments and zero-knowledge (but now only based on OWFs and black-box simulation). Similar

[1] Building on our work, a black-box construction of non-malleable zero-knowledge using OWFs has also recently been obtained by Jain and Pandey [JP11].

[2] Our work is incomparable to the work of Ishai, Prabhakaran and Sahai [IPS08] which gave a constant round black-box MPC protocol based on an *ideal* OT functionality. Our black-box construction is in the plain model but requires stronger cryptographic primitives.

to [Pas04], we obtain constant round construction of multi-party computation (but now only based on constant round OT and black-box simulation). The two slot simulation technique was subsequently used in several others works in order to resolve the issues arising out of mauling attacks (see e.g., [PR05a, MPR06, GJ10]). We believe our techniques maybe able to improve these works as well (although we have not checked the details).

Our constructions and the proofs of security are relatively short and simple. Our primary technique is to have challenge strings of different lengths in the left and the right interaction. A rough initial intuition is as follows. Consider a man-in-the-middle adversary $\mathcal{M}$. In the right interaction, $\mathcal{M}$ is required commit to and then answer a "large" number of randomly generated "puzzles". However in the left interaction, $\mathcal{M}$ is getting a commitment and then an answer to only a "small" number of random puzzles. (This is enforced by having a longer challenge string on the right compared to the one on the left). Thus, it seems intuitive that in the right interaction, $\mathcal{M}$ must be able to compute the answer to a relatively large number of puzzles on its own (without any help from the left interaction).

Non-Aborting Adversaries. To illustrate our idea, we first show how to construct a simple constant round non-malleable commitment scheme for the case of a non-aborting synchronizing[3] adversary. Let the tags in the left and the right interaction be tag and $\tilde{tag}$ respectively. We assume that $tag < \tilde{tag}$ and the tags are of length $\log n$ (the general case can be handled by using the encoding techniques from [DDN91, PR05b]). A sketch of the protocol (described for the left interaction) is given in figure 1.

Tag: Let the tag for the interaction be tag . Let $\ell = k \cdot tag$ (where k denotes the security parameter).

Secret input to the committer: The string ν to be committed

Protocol:

1. The committers $\mathcal{C}$ generates ℓ random strings $r_1, \ldots, r_\ell$ and commits to each of these using a statistically binding commitment scheme.
2. The receiver sends a random challenge $ch \in [\ell]$. (Thus the domain which the challenge string comes from is different for different tags).
3. The committer $\mathcal{C}$ decommits the commitment to the string r_{ch}.
4. The committer now sends a message which allows recovery of ν using any two of the ℓ random strings. In more detail, $\mathcal{C}$ generates ℓ shares of ν using a 2-out-of-ℓ secret sharing scheme. Let these shares be $\nu_1, \ldots, \nu_\ell$. $\mathcal{C}$ now sends $r_1 \oplus \nu_1, \ldots, r_\ell \oplus \nu_\ell$. Finally, $\mathcal{C}$ proves in zero-knowledge that this message is correctly constructed. That is, the values sent are valid secret shares of a single string under a 2-out-of-ℓ secret sharing scheme masked under the random strings committed to in the beginning.

Figure 1: A commitment scheme for non-aborting adversaries

To prove non-malleability of this protocol, we show an extractor that extracts the committed value from the right without rewinding the left interaction at all.

- Since the number of possible challenges in the right interaction is larger than that in the left, by the pigeonhole principle, there must exist two challenges on the right $(\tilde{ch}_1, \tilde{ch}_2)$ s.t. when given either of these, the (synchronizing) man-in-the-middle $\mathcal{M}$ queries with the same challenge ch on the left.
- The extractor executes the first message of the left and the right sessions honestly. It then finds a pair of right challenges $(\tilde{ch}_1, \tilde{ch}_2)$ and the corresponding left challenge ch with the properties discussed above. This is done by rewinding the adversary and observing the left challenges for different right challenges.

[3] Roughly, this means that the man-in-the-middle $\mathcal{M}$ sends the i-th round message on the right immediately after getting the i-th round message in the left interaction.

- The extractor now completes the left and the right sessions honestly by giving the challenge $\tilde{ch}_1$ in the left session.
- The extractor now rewinds the right session and gives the challenge $\tilde{ch}_2$. Note that the left interaction will remain identical and the extractor can simply replay the earlier messages.
- Thus, upon getting the two random strings corresponding to $(\tilde{ch}_1, \tilde{ch}_2)$, the extractor is able to extract the committed value on the right.

Moving to General Adversaries. The above idea does not directly extend to the case of general (i.e., possibly aborting) adversaries. Since the challenge asked comes from only a polynomial-size domain, an aborting adversary can always have a one to one mapping of the challenges on the right to the challenges on the left (and abort on the remaining challenges on the right). Nonetheless, the above idea nicely illustrates the technique of using *challenges coming from domains of different size* in the left and the right interaction (which is the core behind the current work).

To move to the general case, we first modify the above protocol s.t. the receiver sends a challenge chosen from an exponential-size domain (i.e., is a bit string of super-logarithmic length dependent on the size of the tag). Further, the domain of challenges on the right is exponentially larger than the domain of challenges on the left. Hence, there are guaranteed to be such "collisions" as in the case of non-aborting adversaries (unless the adversary only completes the protocol with negligible probability).

The above approach runs into the following problem. Since the challenges are now coming from an exponential size domain, the extractor might not be able to sample such a collision efficiently (even though one exists). The majority of the technical work required in this paper is to resolve this problem. Our key idea is to rely on being able to give a fake "simulated" response to the man-in-the-middle in the left interaction (instead of finding a collision and replaying the same old information). We change the protocol to have the committer simply reveals the committed strings asked for without providing the associated decommitment information. At a high level, our protocol is given in figure 2.

Tag: Let the tag for the interaction be tag . Let $\ell = k \cdot tag$ (where k denotes the security parameter).

Secret input to the committer: The string ν to be committed

Protocol:

1. The committer $\mathcal{C}$ generates ℓ pairs of random strings $\{\alpha_i^0, \alpha_i^1\}_{i \in [\ell]}$ and commits to these using a statistically binding commitment scheme.
2. The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random ℓ-bit challenge string $ch = (ch_1, \ldots, ch_\ell)$.
3. The committer $\mathcal{C}$ sends to $\mathcal{R}$ the values $\alpha_1^{ch_1}, \ldots, \alpha_\ell^{ch_\ell}$. Note that $\mathcal{C}$ *does not* send the openings associated with the corresponding commitments.
4. The committer now sends a message which allows recovery of ν using both strings in any of the ℓ pairs of random strings. Finally, $\mathcal{C}$ proves in zero-knowledge that this message is correctly constructed as well as the values revealed in the previous step were correct.

Figure 2: The skeleton of our non-malleable commitment scheme

The advantage of having an extractor which works without rewinding the left session is that there are no hybrid experiments to consider and one can simply rely on computational hiding property of the commitment scheme to show non-malleability. Our techniques involve the analysis of a basic protocol block which is used naturally in the design of larger cryptographic protocols; hence we believe that our techniques are of independent interest and might be useful elsewhere.

Non-Malleability w.r.t. Replacement. A simple modification of our commitment scheme gives a construction which makes use of the underlying one-way function in a black-box way. The modified construction satisfies the notion of what we call *non-malleability w.r.t. replacement* against synchronizing adversaries. Non-malleability w.r.t. replacement is a slightly weaker yet natural notion of non-malleability which we believe suffices for many application of non-malleable commitments. To get the main idea behind our new notion, consider a man-in-the-middle $\mathcal{M}$ interacting with a committer on the left and a receiver on the right. We consider the right interactions where $\mathcal{M}$ produces an invalid commitment (i.e., a commitment to $\perp$). We now imagine a new "more powerful" adversary $\mathcal{M}$' which behaves exactly like $\mathcal{M}$ except that in some right interactions, whenever the adversary $\mathcal{M}$ would have committed to $\perp$, $\mathcal{M}$' commits to a valid value. We now prove non-malleability w.r.t. this new presumably more powerful adversary (i.e., show that even $\mathcal{M}$' will be unable to make the value in the right interaction dependent on the one in the left).

We show that a commitment scheme which is non-malleable only w.r.t. replacement is sufficient to obtain a (fully) black-box multi-party computation protocol. This allows us to obtain a constant round multi-party computation protocol making only a black-box use of the standard cryptographic primitives with polynomial-time hardness thus directly improving upon the recent work of Wee [Wee10]. Our construction as well the description of the simulator is quite similar to that of Wee [Wee10] (which in turn relies on the works in [IKLP06, CDSMW09]). Our main novelty lies in the analysis of the failure probability of the simulator which in our case only relies on a significantly weaker non-malleability guarantees from the underlying commitment scheme.

Open Problem. An important problem left open by our work is a black-box construction of a commitment scheme *non-malleable w.r.t. commitment.* While considering non-malleability w.r.t. commitment, a number of tools used for getting black-box constructions seem to break down. To our knowledge, getting even a $O(\log n)$ protocol for this problem is currently open.

2 Preliminaries

We follow the definition of non-malleable commitments introduced by Pass and Rosen [PR05a] and further refined by Lin et al [LPV08] (these in turn build on the original definition of Dolev et al [DDN91]). In the real interaction, there is a man-in-the-middle adversary $\mathcal{M}$ interacting with a committer $\mathcal{C}$ (such that the value $\mathcal{C}$ is committing to is ν) in the left session and interacting with a receiver $\mathcal{R}$ in the right session. Let $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ denote a random variable that describes the value $\tilde{\nu}$ the $\mathcal{M}$ commits to the right execution and the view of $\mathcal{M}$ in the full experiment. In the simulated experiment, a simulator $\mathcal{S}$ directly interacts with $\mathcal{R}$. Let $sim^{S}_{\langle C,R\rangle}(1^k, z)$ denote the random variable describing the value $\tilde{\nu}$ committed to by $\mathcal{S}$ and the output view of $\mathcal{S}$. If the tag tag for the left interaction is equal to the tag $\tilde{tag}$ for the right interaction, the value $\tilde{\nu}$ committed to in the right interaction is defined to be $\perp$ in both experiments.

Definition 1 (Non-Malleable Commitments)**.** *A commitment scheme $\langle C, R\rangle$ is said to be non-malleable if for every* PPT *man-in-the-middle adversary $\mathcal{M}$, there exists a* EPPT *simulator $\mathcal{S}$ such that the following ensembles are computationally indistinguishable:*

$$\{mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)\}_{\nu\in\{0,1\}^k, k\in N, z\in\{0,1\}^*} \text{ and } \{sim^{S}_{\langle C,R\rangle}(1^k, z)\}_{\nu\in\{0,1\}^k, k\in N, z\in\{0,1\}^*}$$

Similarly, one can define one-many and many-many variants of the above definition where the view of $\mathcal{M}$ along with the tuple of values it commits to is required to be indistinguishable regardless of the (tuple of) value(s) committed to in the left interactions. We refer the reader to [LPV08] for more details. We also define the notion of one sided (one-one) non-malleable commitments where we only consider interactions where the tag tag for the left interaction is smaller than the tag $\tilde{tag}$ for the right interaction (if $tag \geq \tilde{tag}$, the value $\tilde{\nu}$ committed to in the right interaction is defined to be $\perp$ in both experiments).

Building Blocks. We shall make use of a commitment scheme based on Naor's statistically binding commitment scheme [Nao91] and denote it by com_σ. For simplicity, we assume that this commitment scheme has the property that given the random tape used to construct the commitment, it is possible to recover the committed string as well. This can be achieved by, e.g., simply committing to the string one bit at a time. In addition, we shall make use of a constant round (computational) zero-knowledge argument based on any OWF. We denote such a protocol by ZK . Let k be the security parameter.

3 Construction of Non-Malleable Commitments

In this section, we first describe our basic protocol for "small" tags with one sided non-malleability. They can be extended to the general case by relying on techniques from [PR05b]. We assume that each execution has a tag $tag \in [2n]$. Denote by ℓ the value $k \cdot tag$. Let $\mathrm{com}_\sigma(m)$ denote a commitment to the message m with the first message σ under the statistically binding commitment scheme of Naor. Whenever we need to be explicit about the randomness used to generate the commitment, we denote it as $\mathrm{com}_\sigma(m; r)$ where r is the said randomness. The commitment scheme $\langle C, R\rangle$ between a committer $\mathcal{C}$ trying to commit to ν and a receiver $\mathcal{R}$ proceeds as follows.

Commitment Phase.

0. **Initialization Message.** The receiver $\mathcal{R}$ generates the first message σ of the Naor commitment scheme and sends it to $\mathcal{C}$.

   Primary Slot

1. The committer $\mathcal{C}$ generates ℓ pairs of random strings $\{\alpha_i^0, \alpha_i^1\}_{i\in[\ell]}$ (with length of each string determined by the security parameter). $\mathcal{C}$ further generates commitments of these strings $\{A_i^0 = \mathrm{com}_\sigma(\alpha_i^0), A_i^1 = \mathrm{com}_\sigma(\alpha_i^1)\}_{i\in[\ell]}$ and sends them to $\mathcal{R}$ ($\mathcal{C}$ uses fresh randomness to generate each commitment).
2. The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random ℓ-bit challenge string $ch = (ch_1, \ldots, ch_\ell)$.
3. The committer $\mathcal{C}$ sends to $\mathcal{R}$ the values $\alpha_1^{ch_1}, \ldots, \alpha_\ell^{ch_\ell}$. Note that $\mathcal{C}$ *does not* send the openings associated with the corresponding commitments. $\mathcal{R}$ responds with an acknowledgement message on receiving these values [4].
4. **Verification Message.** Define ℓ strings $\{\alpha_i\}_{i\in[\ell]}$ such that $\alpha_i = \alpha_i^0 \oplus \alpha_i^1$ for all $i \in [\ell]$. $\mathcal{C}$ generates ℓ commitments $B_i = \mathrm{com}_\sigma(\nu; \alpha_i)$ for $i \in [\ell]$ and sends them to $\mathcal{R}$. (That is, randomness α_i is used to generate the i-th commitment to ν).
5. **Consistency Proof.** The committer $\mathcal{C}$ and the receiver $\mathcal{R}$ now engage in a zero-knowledge argument protocol ZK where $\mathcal{C}$ proves to $\mathcal{R}$ that the above commit phase is "valid". That is, there exist values $\hat{\nu}$, $\{\hat{\alpha}_i, \hat{\alpha}_i^0, \hat{\alpha}_i^1\}_{i\in[\ell]}$ such that for all i:
   - $\hat{\alpha}_i^0 \oplus \hat{\alpha}_i^1 = \hat{\alpha}_i$, and,
   - commitments A_i^0 and A_i^1 are valid commitments to the strings $\hat{\alpha}_i^0$ and $\hat{\alpha}_i^1$ respectively under some random tape, and,
   - commitment B_i is a valid commitment to $\hat{\nu}$ under the random tape $\hat{\alpha}_i$.

Decommitment Phase. The committer $\mathcal{C}$ simply reveals the committed value ν and the randomness used in running steps 1 to 4. The receiver $\mathcal{R}$ checks if the messages in the primary slot and the verification message were computed honestly using the revealed randomness. If so, $\mathcal{R}$ takes the value committed to be ν and $\perp$ otherwise.

Lemma 1. *The commitment scheme $\langle C, R\rangle$ is computationally hiding and statistically binding (in the stand alone setting).*

The proof is this lemma is straightforward and a sketch is provided in appendix A.

Theorem 3. *The commitment scheme $\langle C, R\rangle$ is a one sided non-malleable commitment scheme against a synchronizing adversary.*

[4] This is done for technical reasons to ensure that this and the next message by $\mathcal{C}$ are in different rounds of the protocol

Proof. To prove the above theorem, we construct a standalone machine $\mathcal{S}$ such that the ensembles $\{mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)\}_{\nu\in\{0,1\}^k,k\in N,z\in\{0,1\}^*}$ and $\{sim^{S}_{\langle C,R\rangle}(1^k, z)\}_{\nu\in\{0,1\}^k,k\in N,z\in\{0,1\}^*}$ (call these $\mathsf{dist1}$ and $\mathsf{dist2}$ respectively) are computationally indistinguishable. Our $\mathcal{S}$ works as follows. It starts an interaction with $\mathcal{M}$ by simply honestly committing to the value 0 in the left interaction and relaying messages between $\mathcal{M}$ and $\mathcal{R}$ in the right interaction. We claim that such a machine $\mathcal{S}$ satisfies the required property.

Let $r(k) \geq \frac{1}{\text{poly}(k)}$. Towards contraction, assume that there exists a distinguisher $\mathcal{D}$ such that for infinitely many values of k,

$$Pr[\mathcal{D}(\mathsf{dist1}) = 1] - Pr[\mathcal{D}(\mathsf{dist2}) = 1] \geq 2r(k) \tag{1}$$

Fix any such generic k. Now consider the real experiment in which the adversary $\mathcal{M}$ interacts with a committer $\mathcal{C}$ in the left interaction and with a receiver $\mathcal{R}$ in the right one. We shall now show how to construct an extractor $\mathcal{E}$ which takes as input the view of $\mathcal{M}$ in such an experiment and outputs the value $\tilde{\nu}$ committed by $\mathcal{M}$ in the right interaction with probability at least $1 - r(k)$ without rewinding $\mathcal{C}$ (i.e., having access to the value and the random coins used by $\mathcal{C}$ in the left interaction, similar to [DDN91, LPV08]). It is easy to show that this violates the (standalone) computational hiding of the commitment scheme $\langle C, R\rangle$. This is because by equation 1, the value $\tilde{\nu}$ committed by $\mathcal{M}$ on the right in the real experiment is distinguishable with noticeable probability from when the value 0 is being committed to in the left interaction *even* conditioned on the extractor succeeding in outputting $\tilde{\nu}$. Thus, all that remains to show is an extractor that succeeds with probability at least $1 - r(k)$. For simplicity, the failure probability of our extractor (in the analysis below) is conditioned on the event that given the completed main thread, there is exactly one value $\tilde{\nu}$ ($\neq \perp$) consistent with the transcript of the right interaction. By the soundness of the zero-knowledge argument protocol and the statistical binding property of com_σ, this happens happens with all but negligible probability (and hence the failure probability of the extractor increases by at most an additive negligible term in the general case). Without loss of generality, we assume that the adversary $\mathcal{M}$ is deterministic.

Extractor Description and Analysis Let $\ell(k) = k \cdot tag$ and $\tilde{\ell}(k) = k \cdot \tilde{tag}$. Observe that $\tilde{\ell}(k) - \ell(k) \geq k$. Throughout the description below, we denote the messages of the left execution as in the above protocol description; the notation in the right execution is augmented with "tildes" (e.g., A_i^b would refer to a commitment on the left while $\tilde{A}_i^b$ denotes a commitment in the right interaction).

The extractor $\mathcal{E}$ works as follows. It gets as input a transcript of the honestly executed left and right interactions; we refer to this collective interaction transcript as the "main thread". If $\mathcal{M}$ aborts before the main thread was complete, $\mathcal{E}$ simply outputs $\perp$ and halts[5]. Otherwise, $\mathcal{E}$ rewinds $\mathcal{M}$ up to $\frac{k\tilde{\ell}(k)}{r(k)^3}$. For $j \in [\frac{k\tilde{\ell}(k)}{r(k)^3}]$, do the following.

- $\mathcal{E}$ rewinds the right interaction to the beginning of the step 2 of the protocol. It generates a new random challenge $\tilde{ch}[j] \in \{0,1\}^{\tilde{\ell}(k)}$, sends it to $\mathcal{M}$ and receives the challenge $ch[j] \in \{0,1\}^{\ell(k)}$ for the left interaction from $\mathcal{M}$.

- $\mathcal{E}$ now prepares a response to the challenge $ch[j]$ on its own since it is not allowed to rewind the committer $\mathcal{C}$ and make additional queries. Consider the set of commitments on the left already "recovered" in the main thread (i.e., the ℓ commitments A_i^b such that their value was asked by $\mathcal{M}$ and given by $\mathcal{C}$ in the main thread itself). Now, the challenge $ch[j]$ induces a selection of ℓ commitments on the left. Consider any such selected commitment A_i^b. If its value was recovered in the main thread, $\mathcal{E}$ uses that value to prepare the response. Otherwise, $\mathcal{E}$ simply chooses a random string and uses that in the response in place of its value. Such a *simulated response* is sent to $\mathcal{M}$.

- $\mathcal{E}$ receives the response corresponding to $\tilde{ch}[j]$ from $\mathcal{M}$ in the right interaction. If there is an index i s.t. $\tilde{\alpha}_i^0 \oplus \tilde{\alpha}_i^1$ (where one of $(\tilde{\alpha}_i^0, \tilde{\alpha}_i^1)$ was recovered during the main thread while the other received as part of the current response in rewinding j) allows for opening of the commitment $\tilde{B}_i$, $\mathcal{E}$ recovers the committed value $\tilde{\nu}$ from $\tilde{B}_i$. If not, $\mathcal{E}$ goes to the beginning of this loop.

[5] If in an interaction, the parties $\mathcal{C}$ or $\mathcal{R}$ terminate the protocol due to an obvious cheating by $\mathcal{M}$, we also consider it as $\mathcal{M}$ aborting.

If at the end of $\frac{k\tilde{\ell}(k)}{r(k)^3}$ rewindings, $\mathcal{E}$ still was not successful in outputting the value $\tilde{\nu}$ (due to $\mathcal{M}$ aborting or not revealing the correct values for the commitments), it aborts and outputs $\mathsf{Ext\_Fail}$. The fact that $\mathcal{E}$ runs in probabilistic polynomial time is straightforward to prove since $r(k) \geq \frac{1}{\text{poly}(k)}$. We now analyze the probability of $\mathcal{E}$ outputting $\mathsf{Ext\_Fail}$.

Lemma 2. *The probability that the extractor $\mathcal{E}$ outputs $\mathsf{Ext\_Fail}$ is bounded by $r(k)$ for large enough k.*

Proof. We first establish some terminology.

Preliminaries and Notations. We would have three different categories of the main thread (each satisfying a different property) for which the probability (over the random coins used in the rewinds) of $\mathcal{E}$ outputting $\mathsf{Ext\_Fail}$ is noticeable. We will call them "bad" main threads. Keep in mind that $\mathcal{E}$ never output $\mathsf{Ext\_Fail}$ for main threads which are incomplete (i.e., in which $\mathcal{M}$ already aborted). We define a prefix of the main thread as the transcript of the steps 0 and 1 of the left and the right interaction (i.e., up to the stage where $\mathcal{M}$ is waiting for a challenge from the right). For a particular prefix, let p denote the probability that $\mathcal{M}$ completes the real experiment without aborting (i.e., the probability is taken over the random coins used by $\mathcal{C}$ and $\mathcal{R}$ after step 1 of the interaction).

It is convenient to introduce the notion of a *fraction* of main threads. By the fraction of main threads with a particular property being f, we mean the probability that $\mathcal{E}$ receives (a) a *completed* main thread (i.e., $\mathcal{M}$ did not abort), and, (b) the main thread satisfies the property f (over the random coins of the entire experiment). We choose three arbitrary constants C_1, C_2, C_3 such that $\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} \leq \frac{3}{4}$. Note that these constants could in fact be the same and arbitrarily big. However we choose to use three constants for the sake of making the connections between the different parts of proof more clear.

Lemma 3. *The fraction of main threads for which $p < \frac{r(k)}{C_1}$ is bounded by $\frac{r(k)}{C_1}$.*[6] *We call these threads as main threads of type* $\mathsf{bad1}$.

Proof. The proof of this lemma is straight forward. $\Pr\big[$main thread is of type $\mathsf{bad1}\big] \leq \Pr\big[$main thread has a prefix with $p < \frac{r(k)}{C_1}\big] \cdot \Pr\big[$main thread is completed $\mid p < \frac{r(k)}{C_1}\big] \leq 1 \cdot \frac{r(k)}{C_1}$

□

Now for a given main thread, we define the *dependent set of commitment S* as the following subset of commitments in the right interaction. Intuitively, the dependent set of commitments can be thought of as the commitments in the right interaction which were constructed by mauling one of the commitment from the *unrecovered set* of commitments in the left interaction.[7] This means that $\mathcal{M}$ cannot correctly reveal with "good" probability the value of a commitment in the dependent set of commitments unless it gets a correct value of a commitment in the unrecovered set of commitments. Throughout the paper, by the value of a commitment "revealed correctly" by the $\mathcal{M}$, we mean that all possible valid openings of the commitment are only to this revealed value.

Definition 2 (Dependent Set of Commitments)**.** *S is the* dependent set of commitments *of a main thread iff the following two conditions are satisfied. Denote by ch the challenge by $\mathcal{M}$ in the left interaction in this main thread. The probabilities below are over the random coins of the experiment* after *the prefix completion. For every commitment $\tilde{A}_i^b$ in S,*

1. Interesting: *The probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{R}$ AND its value is revealed correctly by $\mathcal{M}$ on the right is at least $\frac{r(k)}{3C_1}$ (for this prefix).* In more detail, consider the experiment where all three parties (i.e., $\mathcal{M}$, the external committer and the receiver) are rewound and the entire experiment is run again (honestly) starting with the same prefix as in the main thread. In this experiment, we require the probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{R}$ AND its value is revealed correctly by $\mathcal{M}$ to be at least $\frac{r(k)}{3C_1}$.

[6] In other words, the probability that the main thread is completed and has $p < \frac{r(k)}{C_1}$ is bounded by $\frac{r(k)}{C_1}$.

[7] Roughly, the unrecovered set of commitments means the set of commitments in the left interaction whose values were not revealed by the committer in the main thread (and hence not known by the extractor)

2. Dependent: *The probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{R}$ AND its value is revealed correctly by $\mathcal{M}$ on the right is less than* $\frac{r(k)}{2C_2\tilde{\ell}(k)}$ conditioned *on the event that the challenge by $\mathcal{M}$ in the left interaction is ch.* In other words, consider the experiment where all three parties (i.e., $\mathcal{M}$, the external committer and the receiver) keep getting rewound and step 2 of the protocol (for both left and right interactions) is run again starting with the same prefix *until $\mathcal{M}$ chooses the same challenge string as in the main thread in the left interaction* (i.e., until a collision w.r.t. the left challenge is obtained). Once that happens, rest of the steps of the protocol are executed to complete the experiment. In this experiment, we require the probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{R}$ AND its value is revealed correctly by $\mathcal{M}$ on the right be less than $\frac{r(k)}{2C_2\tilde{\ell}(k)}$.

Observe that the first probability in the above definition is dependent only on what the prefix in the main thread is, while, the second one depends on the prefix as well as what the left challenge ch appearing in the main thread is. Both these probability values are well defined for a given main thread. Let $|S|$ denote the number of commitments in the set S.

Lemma 4. *Let S be the dependent set of commitment of a main thread. $|S| > \ell + \log^2 k$ for at most a $\frac{r(k)}{C_2} + negl(k)$ fraction of the main threads. Call these threads as main threads of type* bad2.

Intuition. One way of looking at the above lemma is the following. Assume that the extractor had the power of sampling transcripts with the same prefix and having the same left challenge. Then all except for at most $\ell + \log^2 k$ commitments on the right have a "good" probability of being revealed correct by $\mathcal{M}$ (except in $\frac{r(k)}{C_2}$ + negl(k) fraction of the main threads). Hence such an extractor will be successful except for $\frac{r(k)}{C_2}$ + negl(k) fraction of the main threads. At a high level, this is because there are an exponential number of right challenges for each left challenge (on an average) and obtaining a correct response for any two such right challenges enables extraction.

Proof. For a given prefix, consider a set S for a challenge ch such that $|S| > \ell + \log^2 k$. Now consider any random challenge $\tilde{ch}$ given by $\mathcal{R}$ in the right interaction.

- Probability (over choice of $\tilde{ch}$) that the set of commitments selected by $\tilde{ch}$ and the set S are disjoint is at most $\frac{1}{2^{\ell+\log^2 k}}$. In more detail, this probability is either 0 or $\frac{1}{2^{|S|}}$; 0 when S has a pair of "conflicting" commitments $(\tilde{A}_i^0, \tilde{A}_i^1)$ (such that a challenge $\tilde{ch}$ has to select either $\tilde{A}_i^0$ or $\tilde{A}_i^1$) and $\frac{1}{2^{|S|}}$ otherwise since each commitment in S is selected independently with probability $\frac{1}{2}$.
- Now note that there are at most 2^ℓ possibilities for such a set S (for a given prefix) depending upon the choice of challenge $ch \in \{0,1\}^\ell$.
- Taking the union bound over all such sets, we get that the probability that the set of commitments selected by $\tilde{ch}$ is disjoint with *any* such set S (with $|S| > \ell + \log^2 k$) is at most $\frac{2^\ell}{2^{\ell+\log^2 k}} = \text{negl}(k)$.
- Next observe that by the second condition of the definition 2, the probability that for *some* $\tilde{A}_i^b \in S$, $\mathcal{M}$ revealed the correct value in the right interaction in the main thread is bounded by $\frac{r(k)}{C_2}$ (this follows by a union bound over all $\tilde{A}_i^b \in S$ given that $|S|$ cannot exceed $2\tilde{\ell}(k)$).

Now we have the following:

Pr$\big[$main thread is of type bad2$\big]$ $\leq$ Pr$\big[\tilde{ch}$ does not select any commitment in $S\big]$ + Pr$\big[$main thread is completed $|$ $\tilde{ch}$ selects a commitment in $S\big]$

Pr$\big[$main thread is of type bad2$\big]$ $\leq$ negl(k) + Pr$\big[\exists \tilde{A}_i^b \in S$ s.t. $\mathcal{M}$ revealed the correct value of $\tilde{A}_i^b$ in main thread$\big]$

Hence, the fraction of main threads of type bad2 is bounded by $\frac{r(k)}{C_2} + \text{negl}(k)$

□

This lemma shows that there are at most $\ell + \log^2 k$ commitments on the right which are "dependent" on the left commitments whose value $\mathcal{E}$ did not recover in the main thread. However the total number of commitments on the right is $2 \cdot \tilde{\ell} > 2(\ell + \log^2 k)$ (since $\tilde{tag} > tag$). Hence, it seems that there should exists at least one pair of commitments on the right such that $\mathcal{M}$ can correctly compute both the committed values (without asking for values unrecovered in the main thread).

Looking ahead, the intuition for the rest of the proof is as follows. The primary hurdle is in completing the proof is the following. Our PPT extractor will not have the power of sampling transcripts with "collision" (i.e., with the same left challenge). The extractor gets a different challenge from $\mathcal{M}$ (compared to the main thread) while rewinding $\mathcal{M}$ and provides a "simulated" response. We now need to analyze such an experiment. Intuitively, suppose there is a commitment on the right which is revealed correctly with good probability in the "absence" of values from the unrecovered set of commitments (i.e., conditioned on the event when the left challenge in the same as the main thread). Then this means that the right commitment was not formed by "mauling" one of commitments in the unrecovered set. Hence even if a commitment in the unrecovered was given incorrectly, $\mathcal{M}$ hopefully should still reveal that commitment correctly on the right. A formal analysis now follows. We first introduce the following definition.

Definition 3 (Strictly Dependent Set of Commitments)**.** *G is the* strictly dependent set of commitments *for a main thread iff the following two conditions are satisfied. For every commitment $\tilde{A}_i^b$ in G,*

1. Interesting: *The probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{R}$ AND its value is revealed correctly by $\mathcal{M}$ is at least $\frac{r(k)}{3C_1}$ (for this prefix).* This condition is the same as in the definition of dependent set of commitments and refers to the real honest experiment with the given prefix.

2. Strictly Dependent: *The probability that the commitment $\tilde{A}_i^b$ is selected by $\mathcal{E}$ in a rewinding AND its value is revealed correctly by $\mathcal{M}$ on the right is less than $\frac{r(k)^3}{50\tilde{\ell}(k)^2 C_1 C_2 C_3}$.* In more detail, the probability is in the experiment where $\mathcal{M}$ gets random strings in places of left commitment values which remained unrecovered in the main thread.

Observe that the first probability in the above definition is dependent only on what the prefix in the main thread is, while, the second one depends on the prefix as well as the left challenge ch appearing in the main thread (since it refers to the unrecovered set of commitments). We now prove the following lemma.

Lemma 5. *Let S and G respectively be the dependent set and strictly dependent set of commitment of a main thread. $G \not\subseteq S$ for at most $\frac{r(k)}{C_3}$ fraction of the main threads. Call these threads as main threads of type* bad3*.*

Proof. This lemma is proved by relying on the hiding property of the underlying commitment scheme com. We will prove the lemma by contradiction. Assume that for at least a fraction $\frac{r(k)}{C_3}$ of the main threads, there exists a commitment $\tilde{A}_i^b$ in G but not in S. This means the following 3 conditions are true for this main thread (where the probabilities are taken over the random coins of the experiment after the prefix completion). Some informal intuition for how the proof would go is given in comments enclosed under /* and */.

1. Consider the condition of not being in S. This means that the second condition of being in S is not satisfied (since the first condition is the same as that in G). Conditioned on the event that $\mathcal{M}$ does not ask any of the values from the unrecovered set of commitments (i.e., its challenge on the left is ch w.r.t. which S and G are defined), $\mathcal{M}$ reveals the correct value in $\tilde{A}_i^b$ on the right with "large" probability (i.e., at least $\frac{r(k)}{2C_2\tilde{\ell}(k)}$). /* Since the main thread is identically distributed as a random thread having the left challenge ch and the given prefix, this means that the probability of main thread having the correct value of the commitment $\tilde{A}_i^b$ is large. When this event occurs, we show that it is possible to obtain an advantage in breaking the hiding property of the commitment scheme. */

2. Consider the first condition of being in G. Thus, if the values of the commitments in the unrecovered set are given correctly on the left, $\mathcal{M}$ reveals the correct value in $\tilde{A}_i^b$ on the right with "large" probability (i.e., at least $\frac{r(k)}{3C_1}$). /* Assume that the extractor is given from outside a candidate tuple consisting of the values of all the commitments in the unrecovered set. The values given could either all be correctly given or could all be generated at random. Say that the extractor uses these values to construct the simulated response on the left while rewinding. Then if the candidate tuple of values was correct, the correct value of commitment $\tilde{A}_i^b$ is given on the right with large probability. *Hence, it will match with the value observed in the main thread with large probability* assuming that the event described in the previous bullet occurred (i.e., value of $\tilde{A}_i^b$ was obtained correctly in the main thread).*/

3. Consider the second condition of being in G. That is, if the value of the commitments in the unrecovered set are given randomly on the left (i.e., the response is simulated), $\mathcal{M}$ reveals the correct value in $\tilde{A}_i^b$ on the right with "small" probability (i.e., smaller than $\frac{r(k)^3}{50\tilde{\ell}(k)^2 C_1 C_2 C_3}$). /* Thus, if the candidate tuple of the commitment values was generated at random (as opposed to correct), the correct value of the commitment $\tilde{A}_i^b$ is given on the right only with small probability. *Hence, it will match with the value observed in the main thread with small probability* assuming that the event described in the first bullet occurred. Combining this with the observation in the previous bullet, we show that the extractor will be able to obtain an advantage in distinguishing the correct tuple from a random one.*/

We now provide the details. We construct an adversary $\mathcal{A}$ to show that the above conditions violate the (computational) hiding property of the commitment scheme com. Consider the following experiment between the adversary $\mathcal{A}$ and an external challenger *Chal* .

1. $\mathcal{A}$ starts the execution of $\mathcal{M}$ and gives it honestly the messages in the right session. The messages received from $\mathcal{M}$ in the left session are forwarded to *Chal* and its reply is forwarded to $\mathcal{A}$ until the protocol is completed till step 3 (on both left and right interactions).

2. Now the *Chal* provides to $\mathcal{A}$ a total of $M = \frac{25\tilde{\ell}(k)^2 C_1 C_2 C_3}{r(k)^3}$ candidate tuples for the values in the unrecovered set of commitments on the left. Exactly one of the candidate tuples has correct values for all the commitments in the unrecovered set. All the values in the rest of the candidate tuples are generated by *Chal* randomly. The goal of $\mathcal{A}$ would be guess which of the M tuples is the correct one. $\mathcal{A}$ is not allowed any further interaction with *Chal* (and in particular is not allowed to run the protocol beyond step 3).

3. $\mathcal{A}$ now rewinds $\mathcal{M}$ exactly M times. In the i-th rewind, $\mathcal{M}$ gives a challenge $ch[i]$ on the left (if it aborts at any point, we move on the next rewinding). To construct the response, for the commitments in the unrecovered set picked by $ch[i]$, $\mathcal{A}$ uses the values in the i-th candidate tuple. Observe that for exactly one rewind, the response given by $\mathcal{A}$ would be correct and in all other cases, it would be the *simulated* response as given by the extractor $\mathcal{E}$ when it rewinds.

4. $\mathcal{A}$ proceeds as follows. It selects a commitment $\tilde{A}_i^b$ from the right interaction as a guess for a commitment in $G - S$ (if one exists).

5. Now we consider the case where the following happens. In the main thread, the commitment $\tilde{A}_i^b$ was selected by $\mathcal{A}$ and a value $\tilde{\alpha}_i^b$ was received. There is exactly one rewind (say index ind), such that the commitment $\tilde{A}_i^b$ was selected by $\mathcal{A}$ AND a value $\tilde{\alpha}_i^b[ind] = \tilde{\alpha}_i^b$ was received (i.e., the values seen in the main thread and this rewind match). If that is the case, $\mathcal{A}$ outputs the index ind to *Chal* as its guess for the correct value tuple. In all other cases, $\mathcal{A}$ aborts and outputs $\perp$.

We now analyze the success probability of $\mathcal{A}$. Let E denote the event that main thread is of type bad3 and $E1$ denote the event (E AND $\tilde{A}_i^b \in (G - S)$).

Pr$\big[\mathcal{A}$ outputs the correct guess$\big] \geq$ Pr$\big[$E$\big]$ $\cdot$ Pr$\big[E1|E\big]$ $\cdot$ Pr$\big[$correct value $\tilde{\alpha_i^b}$ for $\tilde{A_i^b}$ appears in the main thread $|E1\big]$ $\cdot$ Pr$\big[$correct value $\tilde{\alpha_i^b}$ appears in the rewind with correct response $|E1\big]$ $\cdot$ Pr$\big[$correct value $\tilde{\alpha_i^b}$ does not appear in any rewind with simulated response $|E1\big]$

(Note that the last 3 probability terms are results of experiments run with independent random coins and hence are independent.)

$$Pr\big[\mathcal{A}\text{ outputs the correct guess}\big] \geq \frac{r(k)}{C_3} \cdot \frac{1}{2\tilde{\ell}(k)} \cdot \frac{r(k)}{2C_2\tilde{\ell}(k)} \cdot \frac{r(k)}{3C_1} \cdot \frac{1}{2}$$

(Note that the expected number of times correct value $\tilde{\alpha_i^b}$ appears in simulated responses is $\frac{r(k)^3}{50\tilde{\ell}(k)^2C_1C_2C_3} \cdot (\frac{25\tilde{\ell}(k)^2C_1C_2C_3}{r(k)^3} - 1) < \frac{1}{2}$, hence at least with probability $\frac{1}{2}$, there are 0 such appearances.)

$$Pr\big[\mathcal{A}\text{ outputs the correct guess}\big] \geq \frac{r(k)^3}{24\tilde{\ell}(k)^2C_1C_2C_3} \tag{2}$$

Now we have the following claim. The proof of this claim is straight-forward and a sketch can be found in appendix A.

Claim 1. *In the above experiment, assuming the commitment scheme com is computationally hiding, the probability of any PPT $\mathcal{A}$ outputting the correct guess is bounded by $\frac{r(k)^3}{25\tilde{\ell}(k)^2C_1C_2C_3} + negl(k)$.*

The above claim is in contradiction to the equation 2. This concludes the proof of lemma 5.

□

Concluding the Analysis of the Extractor $\mathcal{E}$. We now conclude the proof of lemma 2. The rest of the proof is quite straightforward. Very roughly, we have already established that there are only a "small" number of commitments on the right (i.e., commitments in set G) which go from being correct with "large" probability (given a correct response on the left) to being correct only with "small" probability (given a simulated response on the left). Thus, there are sufficiently large number of commitments on the right such that given a simulated response, they are revealed correctly by $\mathcal{M}$ (thus implying success for the extractor $\mathcal{E}$). Details follow.

As earlier, for the prefix of the given main thread, let p denote the probability that $\mathcal{M}$ completes the main thread (i.e., the real experiment) without aborting (i.e., the probability is taken over the random coins after step 1). For the given main thread, let q denote the probability of $\mathcal{E}$ succeeding in extracting in a rewinding using a simulated response. Since $\mathcal{E}$ rewinds $\mathcal{M}$ $\frac{k\tilde{\ell}(k)}{r(k)^3}$ times,

$$Pr\big[\mathcal{E}\text{ aborts}\big] \leq p \cdot (1-q)^{\frac{k\tilde{\ell}(k)}{r(k)^3}}$$

(Exact equality may not be satisfied because $\mathcal{M}$ may abort even before prefix completion.) Now this value is noticeable only if $q = o(\frac{r(k)^3}{\tilde{\ell}(k)})$, or, in other words, $q < \frac{r(k)^3}{50\tilde{\ell}(k)}C_1C_2C_3$. Now, Pr$\big[\mathcal{E}$ aborts$\big] \leq$ Pr$\big[$main thread is of type bad1 or bad2 or bad3$\big]$ + Pr$\big[\mathcal{E}$ aborts | main thread is neither of these 3 types$\big]$.

To compute the second term, we first compute q for the main thread. Note that the main thread being not of type bad2 or bad3 implies that $|G| \leq \ell + \log^2 k$ (since $|S| \leq \ell + \log^2 k$ and $G \subseteq S$). Also, since the main thread is not of type bad1, there are at most $O(\log k)$ commitments in the right interaction for which the probability of getting asked on the right (which happens with probability $\frac{1}{2}$) AND revealed correctly by $\mathcal{M}$ is less than $\frac{r(k)}{3C_1}$ (otherwise, it is easy to show that $p < \frac{r(k)}{C_1}$). Or in other words, there are at least $2\tilde{\ell} - \log^2 k$ commitments on the right with probability of getting asked and revealed correctly is at least $\frac{r(k)}{3C_1}$. Out of these, at most $\ell + \log^2 k$ are in G. Hence, (for large enough k) there are at least $\tilde{\ell} + 1$ commitments, or in other words at least one pair of commitments on the right, such that the probability that such a commitment is selected by $\mathcal{E}$ in a rewinding and $\mathcal{M}$ reveals the correct value is at least $\frac{r(k)^3}{50\tilde{\ell}(k)}C_1C_2C_3$. This means for such a main thread, $q \geq \frac{r(k)^3}{50\tilde{\ell}(k)}C_1C_2C_3$. Thus,

$$Pr[\mathcal{E}\text{ aborts}] \leq \frac{r(k)}{C_1} + \frac{r(k)}{C_2} + \frac{r(k)}{C_3} + \text{negl}(k)$$

$$Pr[\mathcal{E}\text{ aborts}] \leq \frac{3}{4}r(k) + \text{negl}(k)$$

This completes the proof.

□

□

Getting Full-Fledged Non-Malleable Commitments. The construction in the previous section can now be extended to get constant round many-many non-malleable commitments for tags of length n against a general adversary (based only one a one way function). This can be done by applications of known techniques. We provide the basic ideas for such an extension here and provide the full details in appendix B.

Getting non-malleable commitments for small tags: We first construct a "two-sided" non-malleable commitment scheme for small tags (i.e., $tag \in [2n]$) against a synchronizing adversary. This can be done very similar to the construction by Pass and Rosen [PR05b]. Denote by $\ell[a]$ the value $k \cdot tag$ and by $\ell[b]$ the value $k \cdot (2n - tag)$. The idea is to have two slots (each representing a rewinding opportunity) such that for exactly one of these slots, the "tag being used on the right" is larger than the one on the left. The extractor will now rewind this slot and extract the value ν. To prove many-many security of the above scheme, we first prove one-many security by simply applying the extractor $\mathcal{E}$ one by one on all sessions on the right and then resort to a general result of Lin et al [LPV08].

Handling tags of length n: A many-many non-malleable commitment scheme for tags of length $\log(n) + 1$ directly leads to a one-one non-malleable commitment scheme for tags of length n using the so called "DDN LOG N trick" [DDN91, LP09]. A construction for many-many non-malleable commitment scheme for tags of length n can also be directly obtained by a single step of non-malleability amplification from [LP09, Wee10].

Security against Non-Synchronizing Adversaries: As is generally the case, once security against synchronizing adversaries is obtained, it is easy to extend it to obtain security even against a non-synchronizing adversary. A general result along these lines has been claimed by Wee [Wee10]. That is, [Wee10] presents a general transformation of non-malleable commitment schemes that are secure against synchronizing adversaries into one that are secure against arbitrary scheduling strategies using one way functions with only an additive constant increase in round complexity. We also provide an alternate direct construction of a non-malleable commitment scheme secure against general adversaries in appendix B.

Obtaining Constant Round Non-Malleable Zero-Knowledge and Multi-Party Computation. The construction of Lin et al [LPTV10] can be instantiated based on any non-malleable commitment scheme[8] and gives rise to a constant round non-malleable zero-knowledge argument protocol assuming only one way functions Constant round non-malleable zero-knowledge constructions can also be obtained by instantiating our commitment scheme using the protocols in [BPS06, GJO10] at the cost of requiring stronger computational assumptions.

Once we obtain a construction of constant round non-malleable zero-knowledge, it can be used to compile a multi-party computation (MPC) protocol secure only against semi-honest adversaries with only a constant multiplicative overhead. Thus, starting with a constant round oblivious transfer protocol and applying the known compilation techniques on the construction of Beaver et al [BMR90] gives rise to a constant round MPC protocol. Thus, this (unconditionally) proves that *the existence of a constant round oblivious transfer protocol is necessary and sufficient to obtain a constant round MPC protocol.*

[8] The LPTV construction actually requires *robust* non-malleable commitments [LP09] which in turn can be constructed from any non-malleable commitment scheme with only a constant multiplicative overhead.

4 Black-Box Construction of Constant Round Multi-Party Computation

Our commitment scheme $\langle C_1, R_1\rangle$ can be modified to so as to use the underlying one-way function (i.e., the commitment scheme com_σ) in a black-box way. This can be done by simply removing the last step of the protocol where the committer gives a zero-knowledge argument of consistency to the receiver. The resulting scheme still satisfies a weaker but natural notion of non-malleability called *non-malleability w.r.t. replacement* against synchronizing adversaries. Next we show that such a commitment scheme is still sufficient to obtain a constant round black-box MPC protocol. We rely on the techniques of Wee [Wee10] (which in turn relies on the works in [IKLP06, CDSMW09]). Our main novelty lies in the analysis of the failure probability of the simulator. Before going further, we introduce our new natural notion of non-malleability called non-malleability w.r.t. replacement.

4.1 Non-Malleable Commitments w.r.t. Replacement

Let the distribution $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ be as before where each element in the support is of the form $(\mathcal{V}, \tilde{\nu})$; $\mathcal{V}$ represents the view of $\mathcal{M}$ and $\tilde{\nu}$ represents the value it commits to in the right interaction (s.t. $\tilde{\nu} = \perp$ if there is either none or more than one possible value consistent with the transcript of the interaction on the right). We say another distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ is *p-compatible* with $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ if the following holds. The distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ is generated by a *replacer* and is the same as $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ except for the following. The replacer takes an element $(\mathcal{V}, \tilde{\nu})$ in the support of $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ and produces an element $(\mathcal{V}, r\tilde{\nu})$ in the support of $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$, such that, whenever $\tilde{\nu}$ (defined by $\mathcal{V}$) is *not* equal to $\perp$, $r\tilde{\nu} = \tilde{\nu}$ except with probability at most p (over the coins of the entire experiment). However if $\tilde{\nu} = \perp$, $r\tilde{\nu}$ could be any arbitrary string. In other words, whenever $\mathcal{M}$ commits to $\perp$ in the real experiment, the corresponding $r\tilde{\nu}$ in $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ may either be $\perp$ or could be any arbitrary string. Note that we do not insist that the replacer run in probabilistic polynomial time. We are now ready to define non-malleable commitments w.r.t. replacement.

Definition 4 (Non-Malleable Commitments w.r.t. Replacement)**.** *A commitment scheme $\langle C, R\rangle$ is said to be non-malleable w.r.t. replacement if for every* PPT *man-in-the-middle adversary $\mathcal{M}$, for every pair of string (ν_1, ν_2), for every $p = \frac{1}{poly(k)}$, there exists distributions $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$ which are p-compatible with $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$ respectively such that these distributions are also computationally indistinguishable:*

$$\{rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)\}_{\nu_1\in\{0,1\}^k, k\in N, z\in\{0,1\}^*} \overset{c}{\equiv} \{rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)\}_{\nu_2\in\{0,1\}^k, k\in N, z\in\{0,1\}^*}$$

A stronger definition would correspond to be p being negligible in the above definition. That is, we require that for every PPT man-in-the-middle adversary $\mathcal{M}$, for every pair of string (ν_1, ν_2), there exists computationally indistinguishable distributions $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$ which are also $\mathrm{negl}(k)$-compatible with $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$ respectively. In this paper, we work with the weaker version described above.

To understand the intuition behind why the notion of non-malleability w.r.t. replacement might be good enough in many application of non-malleable commitments, consider the following two (informally described) experiments:

- This experiment is the real experiment with the real adversary $\mathcal{M}$ (where the distribution of the view and the value committed to by $\mathcal{M}$ is given by $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$).
- This experiment consider an adversary $\mathcal{M}$' which behaves exactly like $\mathcal{M}$ except that in some executions of the commitment protocol, whenever the adversary $\mathcal{M}$ would have committed to $\perp$, $\mathcal{M}$' commits to a valid value. However when asked to open the commitment, $\mathcal{M}$' may simply aborts.

It seems intuitive that the new adversary $\mathcal{M}$' is at least "as powerful" as the adversary $\mathcal{M}$. In any particular protocol execution, a successful attack which can be launched with having an invalid commitment may also be launched by having a valid commitment (but reserving the option of not opening later on). Non-malleability with replacement implies that even the adversary $\mathcal{M}$' (which corresponds to the distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$) is not successful in committing to a value "dependent" on the value in the left interaction.

One can also define one-many variant of the above definition where for all $p = \frac{1}{\text{poly}(k)}$, we require the existence of $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ with the following properties. Note that each element in the support of $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ is of the form $(\mathcal{V}, \tilde{\nu}_1, \ldots, \tilde{\nu}_n)$ while an element in the support $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ is of the form $(\mathcal{V}, r\tilde{\nu}_1, \ldots, r\tilde{\nu}_n)$. We require $\forall i$, $r\tilde{\nu}_i = \tilde{\nu}_i$ if $\tilde{\nu}_i \neq \perp$ except with probability p. Finally, we still require indistinguishability between $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$. We remark that a one-many non-malleable commitment scheme w.r.t. replacement may not necessarily be many-many as well. This is because the replacement strategy (to construct a compatible distribution) may be dependent on the particular left session as well (and hence the strategy could proceed differently for different sessions on the left in case there are multiple such sessions; this would make the usual hybrid argument break down). The notion of non-malleability w.r.t. replacement is very related in spirit to the notion of non-malleability w.r.t. extraction introduced by Wee [Wee10]. Our notion can be seen as a relaxation of the notion of Wee where we allow the extractor to: (a) run in super-polynomial time, and, (b) depend upon the transcript of the left interaction as well.

4.2 Constructing Black-Box Commitments Non-Malleable w.r.t. Replacement

We denote the scheme obtained by removing the last step of the protocol $\langle C_1, R_1\rangle$ (where the committer gives a zero-knowledge argument of consistency to the receiver) to be $\langle bC, bR\rangle$. Note that the scheme $\langle bC, bR\rangle$ makes a black-box use of the underlying commitment scheme com_σ (and hence of the underlying one-way function). Furthermore, this modification does not affect the computational hiding property of the commitment scheme. We now show that the resulting scheme is non-malleable w.r.t. replacement (against synchronizing adversaries).

Lemma 6. *The commitment scheme $\langle bC, bR\rangle$ is a one-many non-malleable commitment scheme w.r.t. replacement against synchronizing adversaries.*

Proof. The proof of this lemma easily follows from the construction of our extractor $\mathcal{E}$ which extracts the committed value from the right interaction without rewinding the left interaction at all. The extractor $\mathcal{E}$ will be used in the construction of a compatible distribution required to prove non-malleability w.r.t. replacement. We provide the proof sketch in what follows. Recall that for every $p = \frac{1}{\text{poly}(k)}$, we need to construct a p-compatible distribution (i.e., with the probability error at most p). The idea would to be use an extractor which fails to extract the correct committed value from $\mathcal{M}$ at most with probability p (conditioned on the commit phase being “valid”). Details follow.

For the given PPT adversary man-in-the-middle adversary $\mathcal{M}$, and $p = \frac{1}{\text{poly}(k)}$, consider the following:

- Run the real experiment with the adversary $\mathcal{M}$. Let $\mathcal{V}$ and $\tilde{\nu}$ be the view of the adversary and the committed value on the right respectively.

- Run the extractor $\mathcal{E}$ with the error parameter p to extract a value in the right interaction. Call the extracted value to be $r\tilde{\nu}$ (and set $r\tilde{\nu}$ to $\perp$ if the extractor was unsuccessful). The resulting $\mathcal{V}$ and $r\tilde{\nu}$ define the distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$.

- The distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$ is p-compatible with $mim^{\mathcal{M}}_{\langle C,R\rangle}(\nu, z)$. This is because $\mathcal{E}$ fails to extract the committed value with probability at most $p(k)$ conditioned on the commit phase being “valid” (i.e., if $\tilde{\nu} \neq \perp$). Hence, $r\tilde{\nu} = \tilde{\nu}$ if $\tilde{\nu} \neq \perp$. $r\tilde{\nu}$ could be an arbitrary string if the commit phase was invalid (which can happen with a noticeable probability now since there is no zero-knowledge argument of consistency). However this is anyway allowed by the definition of p-compatibility.

- Recall that for every noticeable quantity $p(k) = \frac{1}{\text{poly}(k)}$, the extractor $\mathcal{E}$ runs in time polynomial in k without rewinding the left interaction. From this it follows that $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_1, z)$ and $rmim^{\mathcal{M}}_{\langle C,R\rangle}(\nu_2, z)$ are computationally indistinguishable (by the hiding property of the commitment scheme $\langle bC, bR\rangle$).

To prove one-many non-malleability of $\langle bC, bR\rangle$, we construct a p-compatible distribution by simply running the extractor $\mathcal{E}$ one by one on each session on the right. The extractor will be run with the error parameter $\frac{p}{N}$ where N is the number of sessions on the right.

□

The above proof shows that the protocol $\langle bC, bR\rangle$ is a non-malleable commitment scheme w.r.t. replacement. However $\langle bC, bR\rangle$ cannot be shown to be even an standalone extractable commitment scheme. The reason for this is a connection to the black-box impossibility result of Goyal and Jain [GJ10] in the setting of covert computation. Very roughly, the difficulty arises from the inability of the extractor to tell whether or not the adversary $\mathcal{M}$ honestly participated in the main thread on the right (and revealed the values asked for by $\mathcal{R}$ correctly). Thus, if the extractor simply keeps running until if is successful in extracting the committed value on the right, it might not terminate in expected PPT.

To overcome this problem, we remark that one can add a generic standalone extraction phase at the end of any commitment scheme. The resulting protocol $\langle beC, beR\rangle$ proceeds as follows.

1. The committer $\mathcal{C}$ commits to the desired string ν to the receiver $\mathcal{R}$ using the commitment scheme $\langle bC, bR\rangle$. Let r denote the random tape used by $\mathcal{C}$ and let $R = (r, \nu)$.

2. The receiver $\mathcal{R}$ generates the first message σ of the Naor commitment scheme and sends it to $\mathcal{C}$.

3. The committer generates k pairs of random shares $\{R_i^0, R_i^1\}_{i\in[k]}$ of the string R. $\mathcal{C}$ further generates commitments of these shares $\{A_i^0 = \mathrm{com}_\sigma(R_i^0), A_i^1 = \mathrm{com}_\sigma(R_i^1)\}_{i\in[k]}$ and sends them to $\mathcal{R}$.

4. The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random k-bit challenge string $ch = (ch_1, \ldots, ch_k)$.

5. The committer $\mathcal{C}$ sends to $\mathcal{R}$ the values $R_1^{ch_1}, \ldots, R_k^{ch_k}$ and the corresponding decommitments.

During the decommitment phase, $\mathcal{C}$ simply reveals the value ν and the entire random tape used in the commitment phase. $\mathcal{R}$ accepts iff the commitment phase was run honestly using the specified random tape and the value ν. The following lemma is straight-forward to prove.

Lemma 7. *The commitment scheme $\langle beC, beR\rangle$ is a one-many non-malleable commitment scheme w.r.t. replacement against synchronizing adversaries. Furthermore, $\langle beC, beR\rangle$ is a (standalone) extractable commitment scheme.*

4.3 The Multi-Party Computation Protocol

We show how to obtain a constant round (fully) black-box MPC protocol. Our construction as well the description of the simulator is identical to that of Wee [Wee10] (which in turn relies on the techniques developed in [IKLP06, CDSMW09]) while using a version of $\langle bC, bR\rangle$ as the commitment scheme in the appropriate step. The only difference from [Wee10] is in the analysis of the failure probability the simulator where Wee relied on the underlying commitment scheme satisfying the notion of a *many-many non-malleable commitment scheme w.r.t. extraction*. We show that in fact it suffices for the commitment to satisfy the weaker notion of one-many non-malleable commitment scheme w.r.t. replacement. Our main theorem in this section is as follows:

Theorem 4. *There exists a (fully) black-box construction of a constant round secure multi-party computation protocol starting from a constant round OT protocol secure against a malicious sender and a semi-honest receiver (unconditional).*

We briefly recall important parts of the construction and the simulator from [Wee10] (and refer the reader to [Wee10] for full details). The key step in obtaining a constant round black-box MPC protocol is to obtain a constant round multi-party parallel 1-out-of-2 oblivious transfer (OT). A main ingredient we use to construct such an OT protocol is an OT protocol secure against a malicious sender but only a *semi-honest receiver*.

Proposition 1. *[Wee10, PVW08] There exists a (fully) black-box construction of a constant round OT protocol secure against a malicious sender and a semi-honest receiver starting from either of lossy encryption schemes, homomorphic encryption schemes, dense cryptosystems or certifiable enhanced trapdoor permutations.*

The oblivious transfer protocol [CDSMW09, Wee10] Σ works as follows:

Phase I: **Random Tape Generation.** The sender and the receiver execute a (standalone secure) coin tossing protocol, at the end of which, (only) the receiver gets $2k$ random tapes.

Phase II: **Executing Parallel Random OTs.** The sender and the receiver now engage in $2k$ parallel executions of a 1-out-of-two oblivious transfer protocol Π with random inputs. The OT protocol Π used is such that it provides security against a malicious sender and a *semi-honest* receiver (see proposition 1). The $2k$ random tapes used by the receiver in these $2k$ executions are as determined in Phase I.

Phase III: **Cut-and-choose.** The sender and the receiver now engage in a (parallel secure) coin tossing protocol to select a random subset Q having k out of these $2k$ executions. This is where we make use a commitment scheme providing suitable non-malleability guarantees. The coin tossing protocol works as follows:

- The sender chooses a random k bit string q_S and commits to it using the protocol $\langle beC, beR \rangle$.
- The receiver now chooses a random k bit string q_R and sends it to the sender.
- The sender decommits to q_S. The string $q = q_S \oplus q_R$ determines the set Q of k selected executions of Π.

The receiver now decommits to the k appropriate random tapes generated in phase I. Sender checks that the corresponding k random OTs were executed honestly by the receiver and aborts if that is not the case.

Phase IV: **Combiner.** The sender and the receiver now execute an OT combiner protocol on the remaining k random OTs. The protocol results in a single (random) OT which is secure as long as *just one* of the k OTs guarantees security against a malicious receiver while all of them guarantee security against a malicious sender.

We now recall the construction of the simulator $\mathcal{S}$ for the above oblivious transfer construction from [Wee10]. We only provide the high level sketch and refer the reader to [Wee10] for full details.

Simulating Σ executions where only the sender is corrupted. The simulator $\mathcal{S}$ picks a random string q. Define a set Q consisting of k executions of Π based on q. $\mathcal{S}$ proceeds as follows.

- The simulator $\mathcal{S}$ executes Phase I honestly and obtains $2k$ random tapes as in the protocol description.
- Now $\mathcal{S}$ and the adversary (acting as the sender) engage in $2k$ executions of the protocol Π. For an execution $i \in Q$, $\mathcal{S}$ executes the protocol Π correctly as in the protocol description. For $i \notin Q$, $\mathcal{S}$ uses the simulator of the protocol Π to simulate the view of the adversary as well as extract its input.
- The simulator $\mathcal{S}$ now "forces" the value q as the result of the coin flipping protocol in this execution of Σ. In more detail, $\mathcal{S}$ extracts the value q_S using the extractor associated with the commitment scheme $\langle beC, beR \rangle$ and then chooses a value q_R s.t. $q = q_S \oplus q_R$. When the adversary opens its commitment to q_S, $\mathcal{S}$ opens the appropriate k random tapes determined by q.
- $\mathcal{S}$ runs the combiner step and extracts the final (random) input of the corrupted sender since it has already extracted its input in all the remaining k executions of Π.

Simulating Σ executions where only the receiver is corrupted.

- In phase I, $\mathcal{S}$ extracts all the $2k$ random tapes generated for the receiver to be used in the $2k$ executions of Π.
- The simulator $\mathcal{S}$ runs the honest sender algorithm in Phase II and III.

- $\mathcal{S}$ computes an execution index $j \notin Q$ s.t. the receiver played honestly in the j-th execution of Π (with the j-th random tape). If no such j exists, it must be the case that the receiver cheated in exactly k of the $2k$ executions of Π and was able to force the outcome of the coin flipping protocol in step III s.t. Q selects exactly the remaining k executions. $\mathcal{S}$ aborts and output failure at this point. Otherwise, $\mathcal{S}$ extracts the input of the receiver in the final protocol using its input in the j-th execution of Π (which in turn can be obtained using the j-th random tape extracted by $\mathcal{S}$).

Analyzing $\mathcal{S}$. As noted in [Wee10], if the above simulator $\mathcal{S}$ outputs failure only with negligible probability, the output of $\mathcal{S}$ in the ideal execution is indistinguishable from the view of the adversary in the real execution (by relying on the analysis from [IKLP06, CDSMW09]). Wee [Wee10] showed that if the underlying commitment scheme $\langle beC, beR \rangle$ satisfies the notion of a *many-many non-malleable commitment scheme w.r.t. extraction*, $\mathcal{S}$ outputs failure only with negligible probability. We here show that in fact it suffices for $\langle beC, beR \rangle$ to satisfy the weaker notion of one-many non-malleable commitment scheme w.r.t. replacement. Towards contradiction, say that $\mathcal{S}$ outputs failure with a noticeable probability. We consider the following series of hybrid experiments.

Experiment $\mathcal{H}_0$. This hybrid experiment corresponds to the actual simulated execution as described above.

Suppose that in the i-th Σ execution, only the receiver is corrupted and $\mathcal{S}$ outputs failure with a noticeable probability ϵ. That is, in this execution, at the end of phase II, there exists a unique string q^* s.t. if the outcome of coin flipping in phase III is q^*, $\mathcal{S}$ output failure . Furthermore in phase III, $\mathcal{S}$ committed to q_S using $\langle beC, beR \rangle$ and the adversary replies with q_R s.t. $q^* = q_S \oplus q_R$. Keep in mind that $\mathcal{S}$ can efficiently compute q^* at the end of phase II. This can be done by using the $2k$ random tape extracted in phase I and checking the phase II messages of the corrupted receiver against these tapes.

In the subsequent hybrids, the session i will be treated as the "left session" and all the sessions where the adversary acts as the sender as the right sessions. We will then rely on the one-many non-malleability w.r.t. replacement property of $\langle beC, beR \rangle$.

Experiment $\mathcal{H}_1$. In this experiment, we change $\mathcal{S}$ to act as follows. Consider all Σ executions where only the sender is corrupted (i.e., the right sessions). Recall that the simulator extracts the values $(q_S[1], q_S[2], \dots)$ the adversary commits to in phase III in these executions (in order to force the output of the coin flipping protocol in phase III to the desired values). Now in this experiment, instead of extracting these values (to determine the responses $(q_R[1], q_R[2], \dots)$), $\mathcal{S}$ starts using the replacer associated with the commitment scheme $\langle beC, beR \rangle$ (guaranteed by the one-many non-malleable w.r.t. replacement property) to generate a new set of values and treats them as the extracted values. Details follows.

In more detail, $\mathcal{S}$ constructs a one-many adversary $\mathcal{M}$ of the commitment scheme $\langle beC, beR \rangle$ (for this particular prefix of the protocol involving transcripts of phase I and II) by: (a) by treating the commitment in execution i (where $\mathcal{S}$ is supposed to provide a commitment to the adversary) as the left session, and, (b) all the commitments given by the adversary in all other executions of Σ as the right sessions. (Note that all the remaining executions where $\mathcal{S}$ is supposed to provide a commitment to the adversary are executed honestly and inbuilt into the machine $\mathcal{M}$). Now start using the replacer on the adversary $\mathcal{M}$ to generate a $\frac{\epsilon}{2}$-compatible distribution. This allows $\mathcal{S}$ to generate a new set of values $(rq_S[1], rq_S[2], \dots)$ which it treats as the extracted values and uses them to compute the responses $(q_R[1], q_R[2], \dots)$. Note that $\mathcal{S}$ is not necessarily PPT in this experiment because of the usage of the replacer associated with commitment scheme non-malleable w.r.t. replacement (although the specific commitment scheme $\langle beC, beR \rangle$ does come with a PPT replacer).

In this experiment, the probability of $\mathcal{S}$ outputting failure can decrease only by $\frac{\epsilon}{2} + \text{negl(k)}$ because of the $\frac{\epsilon}{2}$-*compatibility* condition satisfied by the output distribution of the replacer. In more detail, consider the point where all the commitments (to q_S) have finished in phase III. If the values committed $(q_S[1], q_S[2], \dots)$ to by the adversary are all valid (i.e., not equal to $\bot$), the sampled values $(rq_S[1], rq_S[2], \dots)$ will be identical except with probability $\frac{\epsilon}{2}$. Hence, the probability of $\mathcal{S}$ outputting failure for such a protocol prefix changes only by $\frac{\epsilon}{2}$. If at least one of the values committed to by the adversary is $\bot$, the adversary will clearly fail and abort in some execution of Σ while opening such a commitment (except with negligible probability). Hence, for such prefixes, $\mathcal{S}$ would have output failure only with negligible probability. Hence we conclude that $\mathcal{S}$ still outputs failure with noticeable probability (at least $\frac{\epsilon}{2} - \text{negl(k)}$) in session i in this experiment.

Experiment $\mathcal{H}_2$. In this experiment, we change $\mathcal{S}$ to act as follows. In phase III, instead of committing to the randomly generated string q_S, $\mathcal{S}$ instead commits to an all 0 string. We argue by the non-malleability with replacement property of $\langle beC, beR\rangle$ that the adversary still replies with an string q_R s.t. $q_S \oplus q_R = q^*$ with noticeable probability. This is sufficient to arrive at a contradiction since in this experiment, the view of the adversary is statistically independent of the (random) string q_S. Here our analysis is similar to [Wee10]. More details follows.

Since $\langle beC, beR\rangle$ is one-many non-malleable w.r.t. replacement, it follows that the associated (possibly super-polynomial time) replacer outputs a tuple of values $(rq_S[1], rq_S[2], \dots)$ which is indistinguishable from the one in the previous hybrid experiment. We can now view the $\mathcal{S}$ as a PPT machine which gets the tuple of values $(rq_S[1], rq_S[2], \dots)$ from an external (unbounded) challenger. If the probability of $\mathcal{S}$ outputting failure changes by a non-negligible amount in this experiment, it follows that using $\mathcal{S}$, it is possible to construct a PPT machine distinguishing the distribution $rmim^{\mathcal{M}}_{\langle C,R\rangle}(q_S, z)$ from $rmim^{\mathcal{M}}_{\langle C,R\rangle}(0^k, z)$ with a noticeable advantage. This completes the proof.

Acknowledgements. We wish to thank Amit Sahai for many useful discussions and suggestions about the presentation of the proof of security. We thank Hoeteck Wee for sharing an early copy of his paper [Wee10] and providing many useful comments on an earlier draft of our work. Thanks to Huijia Lin for pointing out an error in the previous proof of Lemma 6. Thanks also to Abhishek Jain, Rafail Ostrovsky, Omkant Pandey, Alon Rosen and Ivan Visconti for useful discussions.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *FOCS*, pages 345–355. IEEE Computer Society, 2002.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2009.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *STOC*, pages 503–513. ACM, 1990.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *FOCS*, pages 345–354, 2006.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2009.

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

[GJ10] Vipul Goyal and Abhishek Jain. On the round complexity of covert computation. In Leonard J. Schulman, editor, *STOC*, pages 191–200. ACM, 2010.

[GJO10] Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Password-authenticated session-key generation on the internet in the plain model. In Rabin [Rab10], pages 277–294.

[IKLP06] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In Jon M. Kleinberg, editor, *STOC*, pages 99–108. ACM, 2006.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In Wagner [Wag08], pages 572–591.

[JP11] Abhishek Jain and Omkant Pandey. The relative complexity of non-malleable protocols. In *Manuscript*, 2011.

[LP09] Huijia Lin and Rafael Pass. Non-malleability amplification. In Michael Mitzenmacher, editor, *STOC*, pages 189–198. ACM, 2009.

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In Rabin [Rab10], pages 429–446.

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

[MPR06] Silvio Micali, Rafael Pass, and Alon Rosen. Input-indistinguishable computation. In *FOCS*, pages 367–378. IEEE Computer Society, 2006.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. pages 232–241, 2004.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In Wagner [Wag08], pages 57–74.

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *FOCS*, pages 563–572, 2005.

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 533–542. ACM, 2005.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In Wagner [Wag08], pages 554–571.

[PW10] Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitments from sub-exponential one-way functions. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 638–655. Springer, 2010.

[Rab10] Tal Rabin, editor. *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[Wag08] David Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*. Springer, 2008.

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

A Remaining Proofs

Proof Sketch for Lemma 1. The proof is this lemma is straightforward and we only provide a sketch. To prove computational hiding, we consider the following hybrid experiments. We first start simulating the protocol ZK in the final step of the commitment phase. Next, for each $i \in [\ell]$, we replace the commitments $\{A_i^0, A_i^1\}$ to be commitments to random strings (as opposed to shares of the string α_i used later as randomness to generate B_i). Finally, for each $i \in [\ell]$, we change the commitment B_i to be a commitment to a random string (as opposed to a commitment to ν). Hence in the final hybrid, the transcript of the commitment stage contains no information about the value ν being committed to. Statistical binding follows from the statistical binding property of the commitment scheme com.

Proof Sketch for Claim 1. The proof of this claim relies on a straight forward hybrid argument[9]. In the i-th hybrid experiment, in the chosen tuple (out of M tuples) *Chal* keeps the values for the first i unrecovered commitments to be random and the rest correct. In the $\ell(k)$-th hybrid, clearly the probability of $\mathcal{A}$ winning is exactly $\frac{1}{M}$ since the chosen tuple distribution is identical to the rest. Hence, there should exists a hybrid i in which the probability of $\mathcal{A}$ winning changes by a noticeable amount from the last hybrid. Then it can shown that the hiding property of the commitment scheme com can be broken with a noticeable advantage.

[9] Since *Chal* provides just the committed values and not any opening to the commitments, there are no issues related to "selected opening attacks" etc (see [BHY09] and the reference therein)

B Getting Full-Fledged Non-Malleable Commitments

The basic construction can now be extended to get constant round full-fledged non-malleable commitments based only one a one way function. This can be done by an application of known techniques. We provide more details here.

We first construct a full-fledged (i.e., "two" sided) non-malleable commitment scheme for small tags (i.e., $tag \in [2n]$) against a synchronizing adversary. This can be done very similar to the construction by Pass and Rosen [PR05b]. Denote by $\ell[a]$ the value $k \cdot tag$ and by $\ell[b]$ the value $k \cdot (2n - tag)$. The idea is to have two slots (each representing a rewinding opportunity) such that for exactly one of these slots, the "tag being used on the right" is larger than the one on the left. The extractor will now rewind this slot and extract the value ν. The protocol $\langle C_1, R_1 \rangle$ is as follows.

0. **Initialization Message.** The receiver $\mathcal{R}$ generates the first message σ of the Naor commitment scheme and sends it to $\mathcal{C}$.

1. **Primary Slot** a

   (a) The committer $\mathcal{C}$ generates $\ell[a]$ pairs of random strings $\{\alpha_i^0[a], \alpha_i^1[a]\}_{i \in [\ell[a]]}$ (with length of each string determined by the security parameter). $\mathcal{C}$ further generates commitments of these strings $\{A_i^0[a] = \mathrm{com}_\sigma(\alpha_i^0[a]), A_i^1[a] = \mathrm{com}_\sigma(\alpha_i^1[a])\}_{i \in [\ell[a]]}$ and sends them to $\mathcal{R}$ ($\mathcal{C}$ uses fresh randomness to generate each commitment).

   (b) The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random $\ell[a]$-bit challenge string $ch[a] = (ch_1[a], \ldots, ch_{\ell[a]}[a])$.

   (c) The committer $\mathcal{C}$ sends to $\mathcal{R}$ the values $\alpha_1^{ch_1[a]}[a], \ldots, \alpha_{\ell[a]}^{ch_\ell[a]}[a]$. Note that $\mathcal{C}$ *does not* send the openings associated with the corresponding commitments.

2. **Primary Slot** b

   (a) The committer $\mathcal{C}$ generates $\ell[b]$ pairs of random strings $\{\alpha_i^0[b], \alpha_i^1[b]\}_{i \in [\ell[b]]}$ (with length of each string determined by the security parameter). $\mathcal{C}$ further generates commitments of these strings $\{A_i^0[b] = \mathrm{com}_\sigma(\alpha_i^0[b]), A_i^1[b] = \mathrm{com}_\sigma(\alpha_i^1[b])\}_{i \in [\ell[b]]}$ and sends them to $\mathcal{R}$ ($\mathcal{C}$ uses fresh randomness to generate each commitment).

   (b) The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random $\ell[b]$-bit challenge string $ch[b] = (ch_1[b], \ldots, ch_{\ell[b]}[b])$.

   (c) The committer $\mathcal{C}$ sends to $\mathcal{R}$ the values $\alpha_1^{ch_1[b]}[b], \ldots, \alpha_{\ell[b]}^{ch_\ell[b]}[b]$. Note that $\mathcal{C}$ *does not* send the openings associated with the corresponding commitments.

3. **Verification Message.** Define $\ell[a]$ strings $\{\alpha_i[a]\}_{i \in [\ell[a]]}$ such that $\alpha_i[a] = \alpha_i^0[a] \oplus \alpha_i^1[a]$ for all $i \in [\ell[a]]$. $\mathcal{C}$ generates $\ell[a]$ commitments $B_i[a] = \mathrm{com}_\sigma(\nu; \alpha_i[a])$ for $i \in [\ell[a]]$ and sends them to $\mathcal{R}$. (That is, randomness $\alpha_i[a]$ is used to generate the i-th commitment to ν). Similarly compute commitments $B_i[b]$, $i \in [\ell[b]]$ in an analogous way and send them to $\mathcal{R}$.

4. **Consistency Proof.** The committer $\mathcal{C}$ and the receiver $\mathcal{R}$ now engage in a zero-knowledge argument protocol ZK where $\mathcal{C}$ proves to $\mathcal{R}$ that the above commit phase is "valid". That is, both the above primary slots and the verification message are correctly executed with the same value ν.

Decommitment Phase. The committer $\mathcal{C}$ simply reveals the committed value ν and the randomness used in the commitment phase. The receiver $\mathcal{R}$ checks if the commitment phase was run honestly using the above randomness (including making sure its a "valid" commit phase). If so, $\mathcal{R}$ takes the value committed to be ν and $\perp$ otherwise.

Proof Sketch. The proof of security of the above construction remains essentially identical to that of our basic construction. Keep in mind that $\mathcal{M}$ is a synchronizing adversary. Assume that $tag \neq \tilde{tag}$. This means that either $\ell[a] < \tilde{\ell}[a]$ or $\ell[b] < \tilde{\ell}[b]$. In the former case, the extractor $\mathcal{E}$ performs its rewindings for the primary slot a (by giving simulated responses for the challenges of $\mathcal{M}$ on the left). In the latter case, $\mathcal{E}$ rewinds the primary slot b assuming the messages before start of primary slot b as the prefix of the protocol. In both cases, the proof of security (and in particular the proof of all of our 3 key lemmas bounding the fraction of bad main threads) remains essentially identical.

Proving many-many security of the above non-malleable commitment scheme. To prove that our scheme is a many-many or concurrent non-malleable commitment scheme (for tags of length $\log(n) + 1$), we first focus on proving one-many security. There are several right executions with tags $\tilde{tag}_1, \ldots, \tilde{tag}_m$ and a left execution with tag tag. The interesting case is when $\tilde{tag}_i \neq tag$ for all $i \in [m]$. Our idea is to simply apply the extractor $\mathcal{E}$ one by one for all m sessions. More precisely, $\forall i \in [m]$:

- Define a machine $\mathcal{M}_i$ which "emulates" all the right sessions except session i on its own and exposes the i-th session to an outside receiver $\mathcal{R}_i$.
- Run the extractor on the machine $\mathcal{M}_i$ giving it as input the left view as in the main thread and the right view of the i-th session in the main thread.

The probability that the extractor fails can be computed by a union bound over the m right sessions (and can be made smaller than $\frac{1}{\mathsf{poly}(k)}$ for any polynomial function $\mathsf{poly}(k)$ as in the previous section). Following [LPV08], we get that the above construction is also a many-many non-malleable commitment scheme. Hence we get the following lemma.

Lemma 8. *The commitment scheme $\langle C_1, R_1 \rangle$ is a many-many non-malleable commitment scheme against synchronizing adversaries for tags of length $\log(n) + 1$ (i.e., tag $\in [2n]$).*

Handling tags of length n. A many-many non-malleable commitment scheme for tags of length $\log(n)+1$ directly leads to a one-one non-malleable commitment scheme for tags of length n using the so called "DDN LOG N trick" [DDN91, LP09]. A construction for many-many non-malleable commitment scheme for tags of length n can also be directly obtained by a single step of non-malleability amplification from [LP09, Wee10]. In particular, we make a direct use of the following result from [Wee10].

Proposition 2. *(Proposition 3.1 in [Wee10]) Given a one-many commitment scheme $\langle C_1, R_1 \rangle$ for tags of length $\log(n) + 1$ w.r.t. synchronizing adversaries, there exists another one-many (and hence many-many) commitment scheme $\langle C_2, R_2 \rangle$ for tags of length n w.r.t. synchronizing adversaries with only an additive constant increase in the round complexity.*

Security against Non-Synchronizing Adversaries. As is generally the case, once security against synchronizing adversaries is obtained, it is easy to extend it to obtain security even against a non-synchronizing adversary. A general result along these lines has been claimed by Wee [Wee10]. That is, [Wee10] presents a simple and general transformation of non-malleable commitment schemes that are secure against synchronizing adversaries into one that are secure against arbitrary scheduling strategies using one way functions with only an additive constant increase in round complexity. Applying this transformation to the commitment scheme $\langle C_2, R_2 \rangle$ (from proposition 2) yields a constant round non-malleable commitment scheme using only one way functions.

We also provide an alternative direct construction of non-malleable commitment schemes against non-synchronizing adversaries. The protocol is a modification of the commitment scheme $\langle C_1, R_1 \rangle$ (for tags of length $\log(n) + 1$). We first provide some intuition behind the modified protocol. Consider a non-synchronizing adversary $\mathcal{M}$. Our earlier proof (for synchronizing adversaries) runs into problems only when in the left interaction, $\mathcal{M}$ asks for the verification message *before* finishing the two primary slots in the right interaction. In this case, the proof of lemma 5 does not go through. This is since it relies on the inability of an adversary to distinguish between a correct value tuple from an incorrect value tuple for the unrecovered set of commitments. However given the verification message, indeed it is easy to explicitly distinguish the correct value tuple from an incorrect one. Thus to make our proof of security go through, we add additional "secondary slots" each of which represents a rewinding opportunity (borrowing ideas from [LP09]). If $\mathcal{M}$ asks for the verification message on the left *before* finishing the two primary slots on the right (in the main thread), it will be possible to exploit these additional rewinding opportunities on the right (such that $\mathcal{M}$ does not ask for messages in the left interaction while $\mathcal{E}$ is rewinding such slots).

Assume that the zero-knowledge protocol ZK has c_{zk} rounds of interaction between the prover and the verifier. The protocol $\langle C_3, R_3 \rangle$ proceeds as follows.

- **Initialization Message:** Identical to protocol $\langle C_1, R_1 \rangle$.
- **Primary Slot** a**:** Identical to protocol $\langle C_1, R_1 \rangle$.
- **Primary Slot** b**:** Identical to protocol $\langle C_1, R_1 \rangle$.
- $c_{zk} + 1$ **Secondary Slots:** For all $j \in [c_{zk} + 1]$, do the following.
  1. The committer $\mathcal{C}$ generates k pairs of random shares $\{\nu_i^0[j], \nu_i^1[j]\}_{i \in [k]}$ of the string ν (i.e., $\nu = \nu_i^0[j] \oplus \nu_i^1[j]$ for all i). $\mathcal{C}$ further generates commitments of these strings $\{C_i^0[j] = \mathrm{com}_\sigma(\nu_i^0[j]), C_i^1[j] = \mathrm{com}_\sigma(\nu_i^1[j])\}_{i \in [k]}$ and sends them to $\mathcal{R}$.
  2. The receiver $\mathcal{R}$ generates and sends to $\mathcal{C}$ a random k-bit challenge string $ch[j] = (ch_1[j], \ldots, ch_k[j])$.
  3. The committer $\mathcal{C}$ sends to $\mathcal{R}$ the committed shares $\nu_1^{ch_1[j]}[j], \ldots, \nu_k^{ch_k}[j]$ along with the corresponding openings.
- **Verification Message:** Identical to protocol $\langle C_1, R_1 \rangle$.
- **Consistency Proof:** The committer $\mathcal{C}$ and the receiver $\mathcal{R}$ now engage in a zero-knowledge argument protocol ZK where $\mathcal{C}$ proves to $\mathcal{R}$ that the entire commit phase above is "valid". That is, both the primary slots, the $c_{zk} + 1$ secondary slots and the verification messages are correctly executed with the same value ν.

Proof. We consider following two different interleavings in the main thread:

- **Case 1: The verification message in the left interaction appears *before* the end of two primary slots in the right interaction.** This case constitutes the new part of our proof where the secondary slots will be useful. Observe that when this case happens:
  - Since the verification message appears after the secondary slots, *all* the secondary slots in the left interaction are executed (along with the verification message) before the primary slot b finishes on the right.
  - Consider the point where the primary slot b in the right interaction finishes. There are at most c_{zk} message remaining in the left interaction (i.e., message of the ZK protocol) and $c_{zk} + 1$ secondary slots remaining in the right interaction.
  - Hence, there exists at least one secondary slot in the right interaction such that during its execution, there are no message in the left interaction (pigeon-hole principle). Call this the secondary slot j.

  Now our extractor $\mathcal{E}$ will rewind the secondary slot j in the right interaction and extract the value ν by giving a different challenge. If during rewinding, $\mathcal{M}$ aborts or changes the scheduling to ask for a message of the ZK protocol (as opposed to its strategy in the main thread), $\mathcal{E}$ simply rewinds and tries with a different challenge. It is easy to see that the expected number of rewindings required is a constant (observe that $c_{zk} + 1$ is a constant). Alternatively, given any $r(k) = \frac{1}{\text{poly}(k)}$, one can construct an extractor which performs a strict polynomial number of rewinds and succeeds with probability at least $(1 - r(k))$.

- **Case 2: The verification message in the left interaction appears *after* the end of two primary slots in the right interaction.** Our proof for this case is similar to the case for synchronizing adversaries. The only difference is the consideration that the secondary slots (but not the verification message) in the left interaction might now appear before the primary slots in the right interaction finish. However during the rewinds, $\mathcal{E}$ does not have to provide the verification message or the final ZK protocol for consistency (if upon rewinding, $\mathcal{M}$ changes its scheduling to ask for such messages, $\mathcal{E}$ simply rewinds again). Hence during the rewinds, $\mathcal{E}$ simply runs the required secondary slot with the value 0 (as opposed to the real value ν being committed to in the left interaction). If the probability of $\mathcal{E}$ outputting $\mathsf{Ext\_Fail}$ changes by a noticeable amount, one can construct an adversary $\mathcal{A}$ to contradict the computational hiding property of the commitment scheme com_σ. In more detail, in the proof of lemma 5, the challenge *Chal* and $\mathcal{A}$ interact as follows. In addition to interacting with $\mathcal{A}$ to complete a primary slot (and giving candidate tuples), *Chal* now additionally allows interaction in *any polynomial* number of secondary slots as well. Thus, $\mathcal{A}$ can rewind $\mathcal{M}$ successfully M times; each time interacting with *Chal* to complete the secondary slots. Now consider the following two hybrid experiments. In the first hybrid, *Chal* has access to the correct value ν and executes the secondary slots honestly. In that case, essentially the same proof of success of $\mathcal{A}$ goes through. In the second hybrid, *Chal* uses the value 0 (as opposed to ν). By the computational hiding property of scheme com_σ, the success probability of $\mathcal{A}$ cannot change by a noticeable amount in the two hybrid.

□

Thus, the above gives us a non-malleable commitment scheme $\langle C_3, R_3 \rangle$ for non-synchronizing adversaries for tags of length $\log(n) + 1$. Similar to before, it can be shown that $\langle C_3, R_3 \rangle$ is also many-many non-malleable by applying the extractor on each right session one by one. By applying to DDN LOG N trick, we get a one-one non-malleable commitment scheme against non-synchronizing adversaries for tags of length n. A many-many non-malleable commitment scheme for tags of length n against such adversaries can be obtained by applying one step of the non-malleability amplification [LP09]. This gives us the following theorem.

Theorem 5. *There exists a constant round many-many non-malleable commitment scheme using only one way functions.*