Non-Malleable Commitments using Goldreich-Levin List Decoding

Vipul Goyal* Silas Richelson†

Abstract

We give the first construction of three-round non-malleable commitments from the almost minimal assumption of injective one-way functions. Combined with the lower bound of Pass (TCC 2013), our result is almost the best possible w.r.t. standard polynomial-time hardness assumptions (at least w.r.t. black-box reductions). Our results rely on a novel technique which we call *bidirectional Goldreich-Levin extraction*.

Along the way, we also obtain the first rewind secure delayed-input witness indistinguishable (WI) proofs from only injective one-way functions. We also obtain the first construction of a distributionally extractable commitment scheme from injective one-way functions. We believe both of these to be of independent interest. In particular, as a direct corollary of our rewind secure WI construction, we are able to obtain a construction of 3-round promise zero-knowledge from only injective one-way functions.

*Carnegie Mellon University. Email: goyal@cs.cmu.edu. Vipul Goyal is supported in part by NSF grant 1916939, a gift from Ripple, a JP Morgan Faculty Fellowship, and a Cylab seed funding award.

†UC Riverside. Email: silas@cs.ucr.edu. Silas Richelson is supported by UC Lab Fees grant LFR-18-548554. All opinions and statements reported here represent those of the authors.

1 Introduction

The notion of non-malleability is central in cryptographic protocol design. Its objective is to protect against a man-in-the-middle (MIM) attacker who has the power to intercept messages and transform them in order to harm the security in other instantiations of the protocol. Commitment is often used as the paragon example for non-malleable primitives because of its ability to almost "universally" secure higher-level protocols against MIM attacks.

Commitments allow one party, called the committer, to probabilistically map a message m into a string, $\mathrm{Com}(m;r)$, which can be then sent to another party, called the receiver. In the statistically binding variant, the string $\mathrm{Com}(m;r)$ should be *binding*, in that it cannot be later "opened" into a message $m' \neq m$. It should also be *hiding*, meaning that for any pair of messages, m, m', the distributions $\mathrm{Com}(m;r)$ and $\mathrm{Com}(m';r')$ are computationally indistinguishable.

A commitment scheme is said to be *non-malleable* [DDN91] if for every message m, no MIM adversary, intercepting a commitment $\mathrm{Com}(m;r)$ and modifying it at will, is able to efficiently generate a commitment $\mathrm{Com}(\tilde{m};\tilde{r})$ to a related message $\tilde{m}$. Interest in non-malleable commitments is motivated both by the central role that they play in securing protocols under composition (see for example [CLOS02, LPV09]) and by the unfortunate reality that many widely used commitment schemes are actually highly malleable. Indeed, man-in-the-middle (MIM) attacks occur quite naturally when multiple concurrent executions of protocols are allowed, and can be quite devastating.

Beyond protocol composition, non-malleable commitments play a crucial role in designing round efficient secure multi-party computation (see [KOS03, Wee10, Goy11], or more recently, [BGJ$^+$18, HHPV18]), authentication schemes [NSS06], as well as a host of other non-malleable primitives (e.g., coin flipping, zero-knowledge, etc.), and even applications as diverse as position based cryptography [CGMO09]. Beyond cryptography, techniques from non-malleable commitments have found applications in designing non-malleable extractor and codes [CGL16], which in turn were used to obtain a breakthrough in constructing non-malleable extractors [CZ16]. Techniques from non-malleable commitments (and non-malleable zero-knowledge) have also found applications in the realm of hardness amplification: in particular in disproving a "dream version" of Yao's XOR lemma [DJMW12].

The last five years have seen significant progress in understanding the necessity for interaction in non-malleable commitments, in terms of the concrete number of messages required. In particular, Goyal, Richelson, Rosen and Vald [GRRV14] constructed four round non-malleable commitments based on the existence of one-way functions (OWF). Goyal, Pandey and Richelson [GPR16] constructed three round non-malleable commitments using quasi-polynomially hard injective one-way functions. Khurana [Khu17] constructed three round non-malleable commitments by relying on the decisional Diffie-Hellman (DDH) assumption. Pass [Pas13] showed an impossibility for non-malleable commitments using 2 rounds of communication, via a black-box reduction to any "standard" intractability assumption. Recently beautiful works have been able to bypass this lower bound using sub-exponential DDH [KS17], and, using time-locked puzzles [LPS17].

Our question. The lower bound of Pass implies that if one relies on standard polynomial-time hardness assumptions, three rounds is the best possible for non-malleable commitments (at least w.r.t. black-box reductions). The state of art for three or more rounds is represented by several incomparable works: 4 round using injective one-way functions [GRRV14], 3 rounds using quasi-polynomial one-way functions [GPR16], and, 3 rounds using the DDH assumption [Khu17]. In this context, the last remaining natural question is: *what is the minimal cryptographic hardness assumption required for constructing*

3-round non-malleable commitments?

1.1 Our Results.

Our main result is the following

Theorem 1. *There exists a construction of three-round non-malleable commitments from injective one-way functions.*

Note that OWF are necessary to construct commitment schemes. In conjunction with the lower bound of Pass, the above theorem completely settles the question of assumptions and round complexity of non-malleable commitments w.r.t. standard polynomial time hardness assumptions (modulo OWF vs injective OWF).

Our key technical tool is a construction of a 3-round distributionally extractable commitment scheme from injective one-way functions. Roughly speaking, this means there is an extractor so that the extracted message agrees with the committed message *as a distribution*. Though weaker than standard extractability, our scheme does not suffer from "over-extraction", or, "under-extraction", so for all malicious C^*, the chance that the extractor outputs $\perp$ is close to the chance that C^* commits to $\perp$ (please see the next Section for further details; or see our formal definition in Section 2). A similar primitive is constructed in [JKKR17], based on number theoretic assumptions (namely either DDH or QR residuosity). Our techniques are essentially unrelated. We use Goldreich-Levin-type arguments to establish both hiding and extraction properties. We call this technique *bidirectional Goldreich-Levin style extraction*. We believe this to be of independent interest.

A crucial building block we construct and use in our work is a $3-$round, delayed-input rewind secure witness-indistinguishable (WI) proof, also from injective one-way functions. This means that the WI property holds even if the prover is rewound and forced to prove multiple different statements all with a fixed first round message. The delayed-input property requires the prover and the verifier to have access to the input (i.e., the statement, and, in case of the prover, the witness) only in the last round. To our knowledge, the problem of rewind secure WI first appeared in [GRRV14] where it was bypassed by constructing a "weakly" rewind secure scheme where the WI property is guaranteed to hold only with probability $1 - \delta$ (where δ is noticeable). The issue of rewind security for delayed input WI has continued to arise in subsequent works [COSV16b, COSV17a, COSV17b] where it was bypassed using different (and sophisticated) techniques. Very recently, the first construction of a delayed input rewind secure WI was given by Badrinarayanan et al [BGJ$^+$18] by relying on the DDH assumption [1]. No such construction has been from any general assumption in any polynomial number of rounds even in the setting where the prover is rewound only once. We prove the following theorem.

Theorem 2. *Assuming injective one way functions, for every (polynomial) rewinding parameter B, there exists a three round delayed-input witness-indistinguishable argument system with B-rewinding security.*

We also note our non-malleable commitments, and, distributionally extractable commitments also have the delayed input property (i.e., the committer requires the input string only in the last round). This property is sometimes useful while using such commitment schemes in designing larger protocol such as secure multi-party computation.

[1] An earlier ePrint version of [BGJ$^+$18] claimed a construction of delayed input rewind secure WI from only injective OWFs. However the construction was subsequently revised to use DDH.

As a direct consequence of the above theorem, we are able to get a construction of 3-round promise zero-knowledge (ZK) using injective one-way functions. The notion of promise ZK was introduced by Badrinarayanan et. al [BGJ+18] who presented a construction based on the DDH assumption. The source of the DDH assumption was their usage of rewind secure WI based on DDH. Promise ZK is a weakening of zero-knowledge which was used by Badrinarayanan et al in constructing the first 4 round MPC from polynomial time hardness assumptions.

Corollary 2.1. *Assuming injective one way functions, there exists a construction of promise zero-knowledge proofs in 3-rounds.*

Subsequent Work. Our construction of delayed-input rewind secure WI was recently used to obtain a construction 4-round MPC from 4-round oblivious transfer [CCG+19]. All previous constructions relied either on sub-exponential hardness assumptions, or, specific number theoretic assumptions [ACJ17, BGJ+18, HHPV18]. Four rounds of interaction are necessary for MPC w.r.t. black-box simulation.

1.2 Technical Overview

Recently, Goyal, Pandey and Richelson [GPR16] constructed a $3-$round commitment scheme based on one-to-one OWF which is non-malleable against a synchronizing adversary (*i.e.*, an adversary who plays the various rounds of the left and right protocol executions one after the other). The only non-trivial non-synchronizing message scheduling for the adversary is the *sequential scheduling* (*i.e.*, the MIM plays the left session fully, then plays the right session). The basic scheme from GPR fails to defend against a sequential MIM, essentially because their scheme is not extractable. They solved this problem by composing their main scheme with a $3-$round extractable commitment scheme. However, since no such scheme was known from one-to-one OWF, they used a simple scheme based on quasi-polynomially hard one-to-one OWF, and thus their full construction inherits this hardness assumption. Our main technical contribution is the construction of a 3-round commitment scheme from one-to-one OWF with extraction properties which, when composed with the main component of the GPR scheme, gives a non-malleable commitment scheme. The remainder of this technical overview focuses on our construction of this primitive.

The Challenge of $3-$Round Extractable Commitment. Consider the following example commitment scheme: C breaks a message m into a pair of secret shares (s_0, s_1) using XOR secret sharing: $m = s_0 \oplus s_1$. In this way, C prepares k such pairs and commits to each of them, using a non-interactive commitment scheme. The receiver R then chooses one share from each pair at random in the second round, and in the third round, the selected shares are opened by C. This basic scheme exhibbits some extractable properties since the extractor can rewind C and with high probability, recover both the shares for at least one pair. However, this scheme suffers from over-extraction, since a cheating committer C^* might prepare the first pair to XOR to a different message than all of the others (so the committed message is $\perp$), and the extractor might not detect this discrepency, for example if C always aborts instead of opening the first share of the first pair. The extractor will realize that it failed to recover both shares of the first pair, and so will not know whether these shares satisfy $s_0 \oplus s_1 = m$ or not. In the first case, the extractor should output m, in the second $\perp$, and the extractor cannot do anything better than guess. Versions of this problem are called overextraction or underextraction, and commitment schemes which suffer from them are called weakly extractable commitment schemes. Weakly

extractable commitments are not sufficient for proving non-malleability against the sequential MIM, since they do not defend against certain types of "selective $\perp$" attacks. This problem can be overcome by using a zero-knowledge proof of consistency, but this adds an additional round of interaction.

Non-Interactive Commitment based on the Goldreich-Levin Theorem. The starting point for our protocol is Blum's non-interactive commitment scheme [Blu81], built from a one-to-one OWF f with $\lambda-$bit inputs. To commit to a bit m, C samples $\mathbf{x}, \mathbf{r} \leftarrow \{0,1\}^\lambda$, and, sends $\big(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{x}, \mathbf{r} \rangle \oplus m\big)$, where $\langle \mathbf{x}, \mathbf{r} \rangle$ is the inner product mod 2: $\langle \mathbf{x}, \mathbf{r} \rangle = x_1 r_1 \oplus \cdots \oplus x_\lambda r_\lambda$. This commitment scheme is perfectly binding since f is one-to-one. Hiding follows from the Goldreich-Levin theorem [GL89]. The proof of the Goldreich-Levin theorem is key to understanding our protocol. We explain here the artifacts from the proof we will need for this high level overview; more details can be found in Section 2.3.

The core of the proof of the Goldreich-Levin theorem is the "prediction implies inversion" lemma, which says that any prediction algorithm which, given $\mathbf{r}$, can predict $\langle \mathbf{x}, \mathbf{r} \rangle$ with probability noticeably better than guessing, can recover $\mathbf{x}$. Specifically, if $\Pr_{\mathbf{r} \leftarrow \{0,1\}^\lambda}\big[\mathsf{Pred}(\mathbf{r}) = \langle \mathbf{x}, \mathbf{r} \rangle\big] \geq 1/2 + \varepsilon$ holds, then Pred can be used to recover $\mathbf{x}$. This is done by choosing $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow \{0,1\}^\lambda$ for $k = \mathcal{O}\big(\log 1/\varepsilon\big)$, and considering the output of Pred on every string in the set

$$\mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k) := \big\{\mathbf{r}_{S,i} \in \{0,1\}^\lambda : \emptyset \neq S \subset \{1, \ldots, k\}, i \in \{1, \ldots, \lambda\}\big\},$$

where $\mathbf{r}_S := \bigoplus_{\alpha \in S} \mathbf{r}_\alpha$, and $\mathbf{r}_{S,i} := \mathbf{r}_S \oplus \mathbf{e}_i$ where $\mathbf{e}_i$ is the $i-$th unit vector. We refer to the strings in this set as *Goldreich-Levin queries* or GL queries. Notice the number of GL queries is exponential in k. It is important for the proof of the Goldreich-Levin theorem, that Pred performs well on the GL queries. This follows from the fact that GL queries are pairwise independent.

The Starting Protocol. Our starting point is a $3-$round, extractable version of Blum's non-interactive commitment. In this scheme, C sends $f(\mathbf{x})$ in the first round, then R sends $\mathbf{r}$ in the second, then C sends $\langle \mathbf{x}, \mathbf{r} \rangle \oplus m$ in the third. It can be shown that this scheme has good extractability properties. The idea is to have the extractor rewind C and send all GL queries in a GL set. As long as C is committing to $m = 0$ with probability $1/2 + \varepsilon$, (the case when $m = 1$ is similar), the extractor can then run the Goldreich-Levin machinery and recover $\mathbf{x}$, and ultimately, m. The hiding of this protocol is however unclear. Indeed, this protocol is unlikely to satisfy the hiding property since a receiver, given $f(\mathbf{x})$, might be able to recover the first bit x_1 of $\mathbf{x}$. In this case, a cheating R^* could send $\mathbf{r} = \mathbf{e}_1$, so that $\langle \mathbf{x}, \mathbf{r} \rangle = x_1$, the value it knows. Then given C's response, R^* could recover m. So to summarize, it seems as though whichever party controls $\mathbf{r}$, the Goldreich-Levin apparatus can be used to get security against the other party (*i.e.*, if C controls $\mathbf{r}$ then we can prove hiding; if R controls $\mathbf{r}$ we can prove extractability).

Bidirectional Goldreich-Levin. Suppose we could construct a single protocol such that: 1) if C^* is corrupt, then an extractor can rewind C^*, sending $\mathbf{r}$ of its choice; and 2) if R^* is corrupt and breaks hiding via a distinguisher D, then D can be rewound and fed with $\mathbf{r}$'s under the control of an inversion algorithm for the OWF f. Under these ideal circumstances, the Goldreich-Levin theorem could be used in both directions, to prove extractability and hiding. So the main question is how do we design a protocol which gives sufficient control of $\mathbf{r}$ to both sides?

Using Coin Flipping? A natural next idea is to try coin flipping to generate $\mathbf{r}$. For example, consider the protocol where C sends $f(\mathbf{x})$ and $\mathsf{Com}(\mathbf{r}_0)$ in the first round, where $\mathbf{x}, \mathbf{r}_0 \leftarrow \{0,1\}^\lambda$, and Com is

a non-interactive commitment scheme. Then R responds with a uniform $\mathbf{r}_1$. Finally, C opens $\mathbf{r}_0$ and sends $\langle \mathbf{x}, \mathbf{r} \rangle \oplus m$, where $\mathbf{r} = \mathbf{r}_0 \oplus \mathbf{r}_1$. The extraction will continue to work, since, by binding, C^* must open his first commitment to $\mathbf{r}_0$. Thus, the extractor will have full control over $\mathbf{r}$ during rewinding. The proof of hiding, however, still remains unclear. Note, a OWF-inversion algorithm can rewind R^* and commit to a new value of $\mathbf{r}_0$. The problem is that R^* retains full control of $\mathbf{r}_1$ (which may be different in different rewinds). Thus, the inverter will get to see many transcripts with many different values of $\mathbf{r}$, but there seems to be no way to ensure that these $\mathbf{r}$'s are GL queries. This means we cannot harness the Goldreich-Levin theorem to prove hiding.[2]

Another idea is to change the protocol so that C commits to many strings in the first round: C sends $\big(f(\mathbf{x}), \mathrm{Com}(\mathbf{r}_0^1), \ldots, \mathrm{Com}(\mathbf{r}_0^\ell)\big)$ where $\mathbf{x}, \mathbf{r}_0^1, \ldots, \mathbf{r}_0^\ell \leftarrow \{0,1\}^\lambda$; R responds with $\mathbf{r}_1 \leftarrow \{0,1\}^\lambda$ as before; finally C chooses $i \leftarrow [\ell]$ and opens $\mathrm{Com}(\mathbf{r}_0^i)$, and sends $\langle \mathbf{x}, \mathbf{r} \rangle \oplus m$, where $\mathbf{r} = \mathbf{r}_0^i \oplus \mathbf{r}_1$. The extraction can still be made to work much like above. The key point is that since $\ell = \mathsf{poly}(\lambda)$, the extractor can choose $i \in [\ell]$ and only try to extract when C^* opens $\mathrm{Com}(\mathbf{r}_0^i)$; by doing this, the extractor can get full control over $\mathbf{r}$. It seems as though we have made progress towards proving hiding, since now the OWF-inverter can rewind R^* to the beginning of the third round and vary which $\mathrm{Com}(\mathbf{r}_0^i)$ it opens. These $\mathbf{r}_0^i$ are under the inverter's control, so it looks like we use Goldreich-Levin to prove hiding, say setting the $\mathbf{r}_0^i$ to the GL queries in a GL set. The problem with this is that R^*'s advantage in the hiding game might be very small: $\varepsilon \ll 1/\ell$. In particular, it might be the case that R^* only breaks hiding when C opens $\mathrm{Com}(\mathbf{r}_0^1)$, and for all other choices $2 \leq i \leq \ell$, R^* has no advantage. In this case, the $\mathbf{r}$'s obtained by opening $\mathrm{Com}(\mathbf{r}_0^i)$ for $2 \leq i \leq \ell$ will not be useful to the inverter. Thus, this protocol fails as the inverter does not have sufficient control of $\mathbf{r}$ for the Goldreich-Levin proof of hiding to work.

Using Implicit Representation of GL Queries. Note that ℓ GL queries can be implicitly represented by only $\mathcal{O}(\log \ell)$ strings. To take advantage of this, let us change our protocol so that C sends $\big(f(\mathbf{x}), \mathrm{Com}(\mathbf{r}_0^1), \ldots, \mathrm{Com}(\mathbf{r}_0^k)\big)$ in the first round; then R sends $\mathbf{r}_1$ in the second; and C sends $\big(\mathbf{r}_0, \langle \mathbf{x}, \mathbf{r} \rangle \oplus m\big)$ in the third round where $\mathbf{r} = \mathbf{r}_0 \oplus \mathbf{r}_1$; and, in addition, C proves to R in WI that $\mathbf{r}_0 \in \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$. Note that ℓ, the number of options for $\mathbf{r}_0$, is exponential in k, so by choosing $k = \omega(\log \lambda)$, we ensure that $\varepsilon \gg 1/\ell$. This is starting to look like our final protocol. However, two more changes are still required, one each to fix the proofs of hiding and extraction. This brings us to our two novel subroutines.

Using Rewind Secure WI to Prove Hiding. The hiding proof for the protocol so far goes as follows. It is assumed that R^* breaks hiding via a distinguisher D who distinguishes with probability 2ε between commitments to $m = 0$ and $m = 1$. Then R^* and D are used to construct a prediction algorithm Pred which takes $(\mathbf{r}_0, \tau_3)$ as input, and outputs a bit. The prediction algorithm is hardcoded with the first two rounds of the commitment, $\big(f(\mathbf{x}), \mathrm{Com}(\mathbf{r}_0^1), \ldots, \mathrm{Com}(\mathbf{r}_0^k), \mathbf{r}_1, \tau_1, \tau_2\big)$, and a guess for the bit $\langle \mathbf{x}, \mathbf{r}_1 \rangle$. The input pair $(\mathbf{r}_0, \tau_3)$ satisfies $\mathbf{r}_0 \in \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$ and (τ_1, τ_2, τ_3) is a WI transcript proving that $\mathbf{r}_0 \in \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$. The assumption that R^* breaks hiding via D means that with non-negligible probability over the hardcoded values, the following holds:

$$\Pr_{\mathbf{r}_0 \leftarrow \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k), \tau_3} \Big[\mathsf{Pred}(\mathbf{r}_0, \tau_3) = \langle \mathbf{x}, \mathbf{r}_0 \rangle \Big] \geq \frac{1}{2} + \varepsilon.$$

[2] A plausible solution here would be to use a two-sided *simulatable* coin flipping which would allow Inv to fully control the $\mathbf{r}$'s by controlling the outcome of the coin flipping protocol. However, this would require an additional round of interaction.

We then apply the Goldreich-Levin theorem to use Pred to invert the OWF, completing the proof of hiding.

However, the pairwise independence condition which is crucial for the Goldreich-Levin proof to go through is more complicated in this context than usual. This is due to the fact that Pred takes the pair $(\mathbf{r}_0, \tau_3)$ as input, rather than just the string $\mathbf{r}_0$. Pairwise independence in our setting essentially translates to "pairwise independence of two views with a fixed first message". Moreover, our notion is computational; we cannot hope for pairwise independence of views in our cryptographic protocol to hold against an unbounded adversary. We formulate the precise pairwise independence we need for our proof of hiding in Section 3.4. Roughly speaking, the requirement is that if the adversary is given two protocol executions with the same first two messages, it cannot distinguish if the two values of $\mathbf{r}_0$ it sees are of the form $(\mathbf{r}_{S,i}, \mathbf{r}_{T,i})$, or, $(\mathbf{r}_{S,i}, \mathbf{r}_{T,j})$ with $i \neq j$ (recall that the strings in $\mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$ are of the form $\mathbf{r}_{S,i} = \left(\bigoplus_{\alpha \in S} \mathbf{r}_0^\alpha\right) \oplus \mathbf{e}_i$).

Observe that the last round messages from two different executions of our protocol would also contain two different last messages of the WI protocol (with the same first message). In such a setting, all bets regarding the security of a typical WI protocol are off. To solve this problem, we use a *delayed input rewind secure WI* protocol which guarantees witness indistinguishability even if the prover is rewound and forced to prove multiple statements with the same first message. To achieve our computational equivalent of pairwise independence, security under a single rewinding turns out to be enough. As mentioned before, getting such a construction in any number of rounds from injective one-way functions (or from any general assumption) has been an open problem. We resolve problem by constructing a 3-round delayed input rewind secure WI from injective one-way functions for any (polynomial) B s.t. the security holds even if the prover is rewound B times. Our protocol is additionally a proof of knowledge. Our key technique relies on using "two-layers" of MPC in the head [IKOS07] along with a careful combinatorial analysis. More details are given in Section 3.2.

Using Unbounded Polynomial Commitments to Prove Extractability. Recall that the first protocol in the "Using Coin Flipping?" paragraph had a working extractability proof. Also recall that extraction for the second protocol of that section (the one where C commits to $\mathrm{Com}(\mathbf{r}_0^i)$, $i = 1, \ldots, \ell$ in in the first round) reduced to extraction in the first since the number of possible $\mathbf{r}_0$'s was a fixed polynomial. This ceases to hold when we switch to implicitly representing the $\mathbf{r}_0$'s using $(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$ for $k = \omega(\log \lambda)$, and so the proof of extractability stops working. Thus we find ourselves in the seemingly problematic scenerio where we require that ℓ, the number of possible $\mathbf{r}_0$'s, is at most polynomial (so extraction works), and is simultaneously larger than $1/\varepsilon$ for arbitrary non-negligible $\varepsilon > 0$ (so that hiding works). Our final protocol modification resolves this conflict and fixes this problem.

The key idea is to allow k to be chosen dynamically during the protocol. Specifically, we design a kGen sub-protocol which takes two rounds, and outputs a value k to C. The flexibility we require is that 1) for any R^*, a simulator can arrange the output to be such that ℓ (recall that ℓ is exponential in k) is an arbitrarily large polynomial (so $\ell \gg 1/\varepsilon$ can be ensured); and 2) for any C^* and $N = \mathsf{poly}(\lambda)$, the chance that k is so that $\ell > N$ is at most roughly $1/N$. The first and second points, respectively, enable the proofs of hiding and extractability to go through. See Section 3.1 for the description of kGen and the formal security guarantees. Armed with this subroutine, our protocol works as follows.

1. $\mathrm{C} \to \mathrm{R}$**:** C sends $\left(f(\mathbf{x}), \mathrm{Com}(\mathbf{r}_0^1), \ldots, \mathrm{Com}(\mathbf{r}_0^\lambda), \sigma_1, \tau_1\right)$ where $\mathbf{x}, \mathbf{r}_0^\alpha \leftarrow \{0,1\}^\lambda$, σ_1 is the first message of kGen and τ_1 is the first message of delayed-input, rewind secure WI proof.

2. $\mathrm{R} \to \mathrm{C}$**:** R sends $(\mathbf{r}_1, \sigma_2, \tau_2)$ where $\mathbf{r}_1 \leftarrow \{0,1\}^\lambda$, σ_2 and τ_2 are the second messages of their protocols.

3. $\mathrm{C} \to \mathrm{R}$**:** Let k be the output of kGen. C sends $(\mathbf{r}_0, \langle \mathbf{x}, \mathbf{r} \rangle \oplus m, \tau_3)$ where $\mathbf{r} = \mathbf{r}_0 \oplus \mathbf{r}_1$ for a random $\mathbf{r}_0 \leftarrow \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$, and where (τ_1, τ_2, τ_3) proves that $\mathbf{r}_0 \in \mathsf{GL}(\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k)$.

Note that even though C commits to $\mathbf{r}_0^1, \ldots, \mathbf{r}_0^\lambda$ in round 1, only the strings $\mathbf{r}_0^1, \ldots, \mathbf{r}_0^k$ are active (*i.e.*, available for generating $\mathbf{r}_0$ in the third round), where k is the output of kGen.

Final Changes. We mention here that the "final protocol" in the previous section is still a simplification of our actual protocol; some standard changes are required. For example, as written the above protocol is not extractable since m only appears in the third round. This is fixed by committing to m in the first round using non-interactive commitment, and using the above to commit to the decommitment information ω. Another required change is due to the fact that we will need two possible witnesses for the WI proof statements, and so some of the protocol parts above will be run twice, and the proof will ensure the statement holds for one of the parts. Finally, the above protocol allows C to commit to a bit. We support string commitments in our final scheme by repeating the above in parallel for each bit separately. See Section 4 for our formal scheme.

Related works. Given their foundational role in cryptography and beyond, a large body of literature has been dedicated to studying how efficiently non-malleable commitments can be constructed under different assumptions. A long line of work studies the round complexity of non-malleable commitments [DDN91, Bar02, PR05b, PR05a, LP09, PPV08, PW10, Wee10, Goy11, LP11, GLOV12, GRRV14, GPR16, COSV16a, COSV16b, GKS16, KS17, LPS17, Khu17]. A lower bound of Pass [Pas13, KS17] showed the impossibility of two-round non-malleable commitment proven secure w.r.t. a black-box reduction to any "standard" polynomial-time intractability reduction. Thus, three rounds are necessary to get non-malleable commitments from standard polynomial-time hardness assumptions (at least w.r.t. black-box reduction). However the questions of obtaining three round non-malleable commitments from minimal assumptions has remained opened. Relevant to our work, Goyal, Pandey and Richelson [GPR16] gave a 3-round construction of non-malleable commitment scheme from injective one-way functions w.r.t. so called synchronizing adversaries who keep the left and the right execution "in sync". That is, the adversary finishes the i-th round on both the left and the right before beginning the $(i+1)$-th round in either execution. A construction against general adversaries was also presented albeit assuming quasi-poly hard injective one-way functions. Khurana [Khu17] was able to obtain a three round construction against general adversaries using the incomparable DDH assumption even obtaining the stronger notion of concurrent non-malleability. In this paper, our primary goal is to obtain three round non-malleable commitments from minimal (or almost minimal) assumptions.

2 Preliminaries

Throughout, we let λ denote the security parameter, and we write $\mathsf{negl}(\lambda)$ for functions which tend to zero faster than λ^{-c} for any constant c. For probability distributions X and Y, we write $\mathsf{X} \approx_{\mathsf{c}} \mathsf{Y}$ if X and Y are computationally indistinguishable: *i.e.* if for all PPT distinguishers D,

$$\left| \Pr_{x \leftarrow \mathsf{X}}\big(\mathsf{D}(x) = 1\big) - \Pr_{y \leftarrow \mathsf{Y}}\big(\mathsf{D}(y) = 1\big) \right| = \mathsf{negl}(\lambda).$$

For an algorithm $\mathcal{A}$, we denote the running time of $\mathcal{A}$ by $\mathsf{T}_{\mathcal{A}}$.

2.1 Non-Malleable and Distributionally Extractable Commitments

In this section we define commitment schemes (Definition 1), non-malleable commitment schemes (Definition 2), and distributionally extractable commitment schemes (Definition 3), the last being a new notion. All commitment schemes in this work are perfectly binding, so we give definitions only for this case.

Definition 1 (Perfectly Binding Commitment). *Let $\langle \mathrm{C}, \mathrm{R} \rangle$ be a two-phase, two party protocol between a committer C and a receiver R which works as follows. In the commit phase, C uses secret input m and interacts with R who uses no input. Let $c = \mathrm{Com}(m; r)$ denote R's view after the commit phase; let $(m, w) = \mathrm{Decom}(c, m, r)$ denote R's view after the decommit phase, which R either accepts or rejects. We say that $\langle \mathrm{C}, \mathrm{R} \rangle$ is a* perfectly binding commitment scheme *if the following properties hold:*

- **Correctness:** If parties follow the protocol, then $\mathsf{R}(c, m, w) = 1$;
- **Perfect Binding:** For all c and $(m, w), (m', w')$, at most one of $\mathsf{R}(c, m, w)$ and $\mathsf{R}(c, m', w')$ is 1;
- **Hiding:** For all m_0, m_1, $\{\mathrm{Com}(m_0; r)\}_r \approx_c \{\mathrm{Com}(m_1; r)\}_r$.

If, moreover, the commitment scheme consists of a single round from C to R, $\langle \mathrm{C}, \mathrm{R} \rangle$ is called a *non-interactive, perfectly binding commitment scheme*. Such schemes can be constructed from any one-to-one one-way function [Blu81].

The MIM Experiment. Given a commitment scheme $\langle \mathrm{C}, \mathrm{R} \rangle$, the *man-in-the-middle experiment*, refers to the situation where an adversarial M plays two executions of $\langle \mathrm{C}, \mathrm{R} \rangle$, once on the left where he interacts with an honest C, and once on the right where he interacts with an honest R. We call such an adversary a man-in-the-middle (MIM). The output of the experiment consists of two transcripts of $\langle \mathrm{C}, \mathrm{R} \rangle$, and the commitment $\tilde{m}$ inside the right session. The experiment is parameterized by a left commitment message m and a left identity id. Thus, $(\mathbb{T}, \tilde{m}) \leftarrow \mathsf{MIM}^{\mathsf{M}}_{m,\mathsf{id}}$. If the right execution has identity $\tilde{\mathsf{id}} = \mathsf{id}$, the experiment outputs $\perp$ automatically.

Definition 2 (Non-Malleable Commitment). *Let $\langle \mathrm{C}, \mathrm{R} \rangle$ be a perfectly binding commitment scheme. We say that $\langle \mathrm{C}, \mathrm{R} \rangle$ is* non-malleable *if there exists a PPT simulator SIM which, on input id, and given oracle access to M, outputs a transcript-message pair, $(\mathbb{T}, \tilde{m})$ such that for all m:*

$$\{(\mathbb{T}, \tilde{m})\}_{(\mathbb{T},\tilde{m}) \leftarrow \mathsf{MIM}^{\mathsf{M}}_{m,\mathsf{id}}} \approx_{\mathsf{c}} \{(\mathbb{T}, \tilde{m})\}_{(\mathbb{T},\tilde{m}) \leftarrow \mathsf{SIM}^{\mathsf{M}}(\mathsf{id})}.$$

Definition 3 (Distributionally Extractable Commitment). *We say that a perfectly binding commitment scheme $\langle \mathrm{C}, \mathrm{R} \rangle$ is* distributionally extractable *if for all $\varepsilon > 0$, there exists an extractor Ext_ε satisfies the following syntax, running time and extraction guarantees.*

- **Syntax:** *Ext_ε is parametrized by $\varepsilon > 0$, gets oracle access to a possibly unbounded cheating C^*, takes a transcript $\mathbb{T}$ of $\langle \mathrm{C}, \mathrm{R} \rangle$ as input and outputs a message m.*
- **Running Time:** *The running time of Ext_ε is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{C}^*}, 1/\varepsilon\big)$.*
- **Extraction:** *Let $\mathsf{val}^{\mathrm{C}^*}$ and $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}$ denote the distributions which generate a transcript $\mathbb{T}$ by running $\langle \mathrm{C}, \mathrm{R} \rangle$ between an honest R and C^*; then $\mathsf{val}^{\mathrm{C}^*}$ outputs $m = \mathsf{val}(\mathbb{T})$, the committed message inside $\mathbb{T}$; $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}$ outputs $m = \mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$. Then for any cheating, unbounded C^*, $\Delta\big(\mathsf{val}^{\mathrm{C}^*}, \mathsf{Ext}_\varepsilon^{\mathrm{C}^*}\big) \leq \varepsilon$.*

2.2 Delayed-Input Rewind Secure Witness Indistinguishable Proofs

Definition 4 (Delayed-Input Interactive Arguments). *[BGJ+18] An n-round delayed-input interactive protocol (P, V) for deciding a language L is an argument system for L that satisfies the following properties:*

- **Delayed-Input Completeness.** *For every security parameter $\lambda \in \mathbb{N}$, and any $(x, w) \in R_L$ such that $|x| \leq 2^\lambda$,*

$$\Pr[(P, V)(1^\lambda, x, w) = 1] = 1 - \mathsf{negl}(\lambda).$$

  where the probability is over the randomness of P and V. Moreover, the prover's algorithm initially takes as input only 1^λ, and the pair (x, w) is given to P only in the beginning of the n'th round.

- **Delayed-Input Soundness.** *For any PPT cheating prover P^* that chooses x^* (adaptively)* after *the first $n - 1$ rounds, it holds that if $x^* \notin L$ then*

$$\Pr[(P^*, V)(1^\lambda, x^*) = 1] \leq \epsilon.$$

  where the probability is over the random coins of V, and, ϵ is known as the soundness error of the protocol.

The next definition is from [BGJ+18] where such a primitive was constructed assuming the polynomial hardness of DDH.

Definition 5 (3-Round Delayed-Input WI with Bounded Rewinding Security). *[BGJ+18] Fix a positive integer B. A delayed-input 3-round interactive argument (as defined in Definition 4) for an NP language L, with an NP relation R_L is said to be WI with B-*Rewinding Security *if for every non-uniform PPT interactive Turing Machine V^*, it holds that $\{\mathsf{REAL}_0^{V^*}(1^\lambda)\}_\lambda$ and $\{\mathsf{REAL}_1^{V^*}(1^\lambda)\}_\lambda$ are computationally indistinguishable, where for $b \in \{0, 1\}$ the random variable $\mathsf{REAL}_b^{V^*}(1^\lambda)$ is defined via the following experiment. In what follows we denote by P_1 the prover's algorithm in the first round, and similarly we denote by P_3 its algorithm in the third round.*

Experiment $\mathsf{REAL}_b^{V^*}(1^\lambda)$:

1. *Run $P_1(1^\lambda)$ and denote its output by (rwi_1, σ), where σ is its secret state, and rwi_1 is the message to be sent to the verifier.*
2. *Run the verifier $V^*(1^\lambda, \mathsf{rwi}_1)$, who outputs $\{(x^i, w^i)\}_{i\in[B-1]}$, x^B, w_0^B, w_1^B and a set of messages $\{\mathsf{rwi}_2^i\}_{i\in[B]}$.*
3. *For each $i \in [B-1]$, run $P_3(\sigma, \mathsf{rwi}_2^i, x^i, w^i)$, and for $i = B$, run $P_3(\sigma, \mathsf{rwi}_2^i, x^i, w_b^i)$ where P_3 is the (honest) prover's algorithm for generating the third message of the WI protocol. Send the resulting messages $\{\mathsf{rwi}_3^i\}_{i\in[B]}$ to V^*.*

In Section 3.2, we construct three-round delayed-input WI with bounded-rewinding security from any one-to-one one-way function for any fixed polynomial rewinding parameter B. Our construction will use as a building block the 3-round delayed-input WI protocol of [LS90] (*i.e.*, the case of $B = 1$ above).

MPC-in-the-Head [IKOS07]. As in [BGJ+18], we make black-box use of a 3-round zero knowledge protocol (non delayed-input) with bounded rewinding security. The soundness error of the protocol would depend upon the rewinding parameter B.

Definition 6 (3-Round ZK with Bounded Rewinding Security). *[BGJ+18] Fix a positive integer B. A delayed-input 3-round interactive argument (as defined in Definition 4) for an NP language L, with an NP relation R_L is said to have B-*Rewinding Security *if there exists a simulator* Sim *such that for every non-uniform PPT interactive Turing Machine V^* and $(x, w) \in R_L$, it holds that $\{\mathsf{REAL}^{V^*}(1^\lambda, x, w)\}_\lambda$ and $\{\mathsf{IDEAL}^{V^*}(1^\lambda, x)\}_\lambda$ are computationally indistinguishable, where the random variable $\mathsf{REAL}^{V^*}(1^\lambda, x, w)$ is defined via the following experiment; $\mathsf{IDEAL}^{V^*}(1^\lambda, x)$ is the output of $\mathsf{Sim}^{V^*}(1^\lambda, x)$.*

Experiment $\mathsf{REAL}^{V^*}(1^\lambda)$**:** *Let P_1/P_3 denote the prover's algorithm in the first/third round.*

1. *Run $P_1(1^\lambda, x, w; r)$ and obtain output rwi_1 to be sent to the verifier.*
2. *Run the verifier $V^*(1^\lambda, \mathsf{rwi}_1)$ and interpret its output as message rwi_2.*
3. *Run $P_3(1^\lambda, \mathsf{rwi}_2, x, w; r)$, where P_3 is the (honest) prover's algorithm for generating the third message of the WI protocol, and send its output rwi_3 to V^*.*
4. *Set a counter $i = 0$.*
5. *If $i < B$, then set $i = i + 1$, and V^* (given all the information so far) generates another message rwi_2^i, and receives the (honest) prover's message $P_3(\mathsf{rwi}_2^i, x, w; r)$. Repeat this step until $i = B$.*
6. *The output of the experiment is the view of V^*.*

Imported Theorem 1. *[IKOS07, BGJ+18] Assume the existence of injective one-way functions. Then, for any (polynomial) rewinding parameter B, there exists a 3-round zero-knowledge protocol for proving NP statements that is simulatable under B-bounded rewinding according to 6.*

If B is a constant, the soundness error of the above protocol will be a constant. If $B = \text{poly}(\lambda)$, the soundness error $\epsilon \leq 1 - q(\lambda)$ where q is also a polynomial.

2.3 The Goldreich-Levin Theorem

An influential result of Goldreich and Levin [GL89] says that every one-way function has a hardcore predicate. The core of their proof is the "prediction implies inversion" lemma, which says roughly that if an algorithm can, given $f(\mathbf{x})$, predict random inner products of $\mathbf{x}$ with probability noticeably better than $1/2$, then this prediction algorithm can be used to invert f and recover $\mathbf{x}$. We will make frequent use of this lemma in our security proofs. We set some notations, and then prove the lemma we need.

Given $\lambda-$bit strings $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \{0,1\}^\lambda$ and a subset $S \subset \{1, \ldots, k\}$ we let $\mathbf{r}_S := \bigoplus_{i \in S} \mathbf{r}_i$. Given S and $i \in \{1, \ldots, \lambda\}$ we let $\mathbf{r}_{S,i} := \mathbf{r}_S \oplus \mathbf{e}_i$ where $\mathbf{e}_i$ is the $i-$th unit vector.

Definition 7. *Given $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \{0,1\}^\lambda$, let*

$$\mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k) := \left\{\mathbf{r}_{S,i} \in \{0,1\}^\lambda : \emptyset \neq S \subset \{1, \ldots, k\}, i \in \{1, \ldots, \lambda\}\right\}.$$

Rackoff's combinatorial proof of the Goldreich-Levin theorem considers sets of the form $\mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ and shows how to recover $\mathbf{x}$ using an algorithm whose prediction success on $\mathbf{r} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ satisfies certain statistical properties. This technique is demonstrated in the proof below.

Definition 8. *Fix $\varepsilon > 0$ and a secret $\mathbf{x} \in \{0,1\}^\lambda$. A* Goldreich-Levin Prediction Algorithm with secret $\mathbf{x}$ and advantage ε *(or just GL-predictor for short), is a randomized procedure* Pred *which takes $\mathbf{r} \in \{0,1\}^\lambda$ as input, and outputs a value in $\{0,1\} \cup \{\perp\}$ such that:*

$$\left| \Pr_{\mathbf{r} \leftarrow \{0,1\}^\lambda} \left[\mathsf{Pred}(\mathbf{r}) = \langle \mathbf{r}, \mathbf{x} \rangle \right] - \frac{1}{2} \right| \geq \varepsilon.$$

Lemma 1. *Let $\mathbf{y} = f(\mathbf{x})$ for a one-to-one function $f : \{0,1\}^\lambda \to \{0,1\}^{\mathsf{poly}(\lambda)}$. Let* Pred *be a GL-predictor with secret $\mathbf{x} \in \{0,1\}^\lambda$ and advantage $\varepsilon > 0$. Then there exists an inversion algorithm* Inv *which, given $\mathbf{y}$ and oracle access to* Pred*, outputs $\mathbf{x}$ with high probability $1 - 2^{-\Omega(\lambda)}$. The running time of* Inv *is $\mathsf{T}_{\mathsf{Inv}} = \mathsf{poly}\big(\lambda, 1/\varepsilon, \mathsf{T}_{\mathsf{Pred}}\big)$.*

Definition 9. *We call the algorithm* Inv *guaranteed by Lemma 1 the* GL-inversion algorithm corresponding to Pred.

Proof of Lemma 1. We give the combinatorial proof due to Rackoff. We assume that the quantity inside the absolute value is positive. This is without loss of generality since Inv can perform the following procedure twice, one time negating all outputs of Pred ensuring the quantity is positive once. Upon producing $\mathbf{x}_1$ and $\mathbf{x}_2$ in this way, Inv outputs the $\mathbf{x}_i$ for which $f(\mathbf{x}_i) = \mathbf{y}$; there can be at most one as f is one-to-one. The Inv procedure we describe below outputs $\mathbf{x}$ with probability at least $\varepsilon^3/16$. This is amplified to $1 - 2^{-\Omega(\lambda)}$ by repeating $\varepsilon^{-3} \cdot \lambda$ times.

Let $k = 3 + 3\log\big(1/\varepsilon\big)$. Inv chooses $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow \{0,1\}^\lambda$ and $b_1, \ldots, b_k \leftarrow \{0,1\}$ at random. For $S \subset [k]$, let $b_S := \oplus_{i \in S} b_i$. With probability $2^{-k} = \varepsilon^3/8$, $b_i = \langle \mathbf{r}_i, \mathbf{x} \rangle$ for all i, in which case $b_S = \langle \mathbf{r}_S, \mathbf{x} \rangle$ for all $S \subset [k]$. For all $\mathbf{r}_{S,i} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$, Inv sets a guess for the $i-$th bit of $\mathbf{x}$, $x_{S,i} = \mathsf{Pred}(\mathbf{r}_{S,i}) \oplus b_S$, and $x_i = \mathsf{MAJORITY}\big(\{x_{S,i}\}_S\big)$. Whenever $b_i = \langle \mathbf{r}_i, \mathbf{x} \rangle$ holds for all $i \in [k]$, the expected number of non-empty $S \subset [k]$ for which $\mathsf{Pred}(\mathbf{r}_{S,i}) = \langle \mathbf{r}_{S,i}, \mathbf{x} \rangle$ is at least $(2^k - 1) \cdot \varepsilon \cdot (1/2 + \varepsilon)$, while the expected number of S for which $\mathsf{Pred}(\mathbf{r}_{S,i}) \neq \langle \mathbf{r}_{S,i}, \mathbf{x} \rangle$ is at most $(2^k - 1) \cdot \varepsilon \cdot (1/2 - \varepsilon)$ (expectation over $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow \{0,1\}^\lambda$). Since the strings $\{\mathbf{r}_{S,i}\}_S$ are pairwise independent for all $i \in [\lambda]$, the probability that x_i is not the $i-$th bit of $\mathbf{x}$, can be bounded using Chebyshev's inequality:

$$\begin{aligned} \Pr\Big[x_i \text{ wrong}\Big] &\leq \Pr\Big[\#\{S : x_{S,i} \text{ cor.}\} \leq (2^k - 1)\varepsilon/2\Big] + \Pr\Big[\#\{S : x_{S,i} \text{ inc.}\} \geq (2^k - 1)\varepsilon/2\Big] \\ &\leq \frac{\varepsilon^{-3}}{2^k - 1} + \frac{\varepsilon^{-3}}{2^k - 1} \leq \frac{1}{2\lambda}. \end{aligned}$$

Thus, $\Pr\big[\exists\, i \text{ st } x_i \text{ wrong} \big| b_i = \langle \mathbf{r}_i, \mathbf{x} \rangle\ \forall\, i\big] \leq 1/2$ follows from the union bound. So conditioned on $b_i = \langle \mathbf{r}_i, \mathbf{x} \rangle$ for all i, Inv recovers all x_i correctly with probability at least $1/2$. The result follows. □

3 Building Blocks

3.1 Unbounded Polynomial Commitment

Here we present a simple, yet key component of our main construction. It is a two round commitment scheme where C commits to an integer. If executed honestly, the committed value is 1 with high

probability. Moreover, even if C cheats, the committed value is at most N with probability proportional to $1/N$. We call this protocol an *unbounded polynomial commitment* because a simulator who is able to rewind R can, in time $\mathsf{poly}(N)$, produce an indistinguishable transcript where the committed value is N. The protocol is the following.

1. $\mathrm{C} \to \mathrm{R}$**:** send $c = \mathrm{Com}(s \circ N; \eta)$ where $s, N \leftarrow \left[2^\lambda\right]$ and $\eta \leftarrow \$$;

2. $\mathrm{R} \to \mathrm{C}$**:** draw and send $s' \leftarrow \left[2^\lambda\right]$;

Committed Value: the committed value is N if $s + s' \equiv 0 \pmod N$; 1 otherwise.

Note that if C sends $c = \mathrm{Com}(s, N; \eta)$ in round 1, then

$$1/2N \leq \Pr_{s' \leftarrow [2^\lambda]}\left[s + s' \equiv 0 \pmod N\right] \leq 2/N. \tag{1}$$

It follows from (1) (upper bound) that a) if C and R play honestly, then the committed value is 1 with probability $1 - 2^{-\Omega(\lambda)}$; b) no matter how C deviates from the protocol, if R plays honestly then the committed value is at most N with probability at least $1 - 2/N$.

3.2 Rewind Secure Delayed-Input Witness-Indistinguishable Proof

Building Blocks. Our construction will make use of two building blocks: the 3-round delayed-input WI protocol in [LS90], and, the bounded rewinding secure 3-round "MPC in the head" based 3-round protocol of [IKOS07].

Theorem 3. *Assuming injective one-way functions, for every (polynomial) rewinding parameter B, there exists a three round delayed-input witness-indistinguishable proof system* RWI *with B-rewinding security.*

The soundness of our protocol depends upon the rewinding parameter B and can be amplified via parallel repetition while preserving the WI property. Our protocol RWI will consist of 4 algorithms $(\mathsf{RWI}_1, \mathsf{RWI}_2, \mathsf{RWI}_3, \mathsf{RWI}_4)$ where the first 3 denote the algorithms used by the prover and verifier to send their messages and the last is the final verification algorithm. We use the protocol from [IKOS07]. We denote its algorithms by $\mathsf{Head.ZK} = (\mathsf{Head.ZK}_1, \mathsf{Head.ZK}_2, \mathsf{Head.ZK}_3, \mathsf{Head.ZK}_4)$, where the first 3 denote the algorithms used by the prover and verifier to generate their messages, and the last is the final verification algorithm. The simulator of the protocol $\mathsf{Head.ZK}$ is denoted by S_{zk}. We will also use the delayed-input WI protocol from [LS90] and denote its algorithms by $\mathsf{DIWI} = (\mathsf{DWI}_1, \mathsf{DWI}_2, \mathsf{DWI}_3, \mathsf{DWI}_4)$, where the first 3 denote the algorithms used by the prover and verifier to generate their messages, and the last is the final verification algorithm.

Let λ be the statistical security parameter. We define parameter $N = B^2\lambda^4$.

3.3 Security Analysis of RWI

Proving Soundness. We prove that our protocol RWI has soundness $\delta/2$ where δ is the soundness parameter of the $\mathsf{Head.ZK}$ construction. Suppose $x \notin L$. Consider the following two cases:

1. **Case 1:** There exists $i \in [N]$ s.t. $c_i \neq \mathsf{Com}(0)$. In this case, we claim that V will reject the execution with probability at least $\delta/2$. This is because with probability $1/2$, the challenge ch will be 0. If so, by the soundness of $\mathsf{Head.ZK}$, V is guaranteed to reject the execution with probability at least δ.

Inputs: At the beginning of the third round, the prover P gets as input (x, w); V gets only x.

1. **Round 1: Prover message:**
   - P prepares and sends commitments $c_1, \ldots, c_N$ where $c_i = \mathsf{Com}(0)$ for all i.
   - P also prepares and sends a first round message $\mathsf{hzk}_1^{P\to V}$ for a single instance of $\mathsf{Head.ZK}$, using $\mathsf{Head.ZK}_1$. The statement for $\mathsf{Head.ZK}$ is that each $c_i, i \in [N]$ is indeed a commitment to 0; P uses the commitment openings as its witness.
   - P also prepares and sends first round messages $\{\mathsf{dwi}_{1,i}^{P\to V}\}_{i\in[N]}$ for N separate instances of DIWI. The statements for these DIWI instances will come in the third round.
2. **Round 2: Verifier message:**
   - The verifier samples a challenge bit ch and sends it to P.
   - If $ch = 0$, V in addition executes $\mathsf{Head.ZK}_2$ to sample $\mathsf{hzk}_2^{V\to P}$ and sends it to V.
   - If $ch = 1$, V executes DWI_2 on $\{\mathsf{dwi}_{1,i}^{P\to V}\}_{i\in[N]}$ to get $\{\mathsf{dwi}_{2,i}^{V\to P}\}_{i\in[N]}$ and sends to P.
3. **Round 3: Prover message:**

   If $ch = 0$, P generates $\mathsf{hzk}_3^{P\to V}$ by running $\mathsf{Head.ZK}_3$ and sends it to P. If $ch = 1$, P proceeds as follows:
   - Following [IKOS07], emulate an MPC computation of the circuit representing the witness relation with λ players. The input of each player will be a share of the witness w. Let the view of the i-th player be V_i. For $i \in [\lambda]$, compute $cv_i = \mathsf{Com}(V_i)$ and send it to V.
   - Select a set of $\lambda(\lambda-1)$ distinct random indices $\{k_{i,j} \in [N]\}_{i\neq j, i\in[\lambda], j\in[\lambda]}$. Represent these set of indices by SI and send them to V.
   - Use $\{\mathsf{dwi}_{1,i}^{P\to V}, \mathsf{dwi}_{2,i}^{V\to P}\}_{i\in SI}$ and the algorithm DWI_3 to generate $\{\mathsf{dwi}_{3,i}^{P\to V}\}_{i\in SI}$ and send them to V. For each $k_{i,j} \in SI$, the message $\mathsf{dwi}_{3,k_{i,j}}^{P\to V}$ prove that either (a) $c_{k_{i,j}}$ is a commitment to 1, or, (b) the views (V_i, V_j) are honest and "consistent" with each other. That is, there exist input (w_i, r_i) (resp (w_j, r_j)) s.t. V_i (resp. V_j) is computed and committed honestly using (w_i, r_i) (resp (w_j, r_j)). Furthermore, each outgoing message sent to the j-th player in V_i is consistent with each incoming message from the i-th player in V_j, and, vice-versa. The honest prover P uses the witness corresponding to (b) to compute $\mathsf{dwi}_{3,k_{i,j}}^{P\to V}$. Note that unlike [IKOS07], there is *no* challenge from the verifier selecting a random subset of the views.
4. **Verifier Output:**
   - If $ch = 0$, compute the output of the algorithm $\mathsf{Head.ZK}_4$ on $(\mathsf{hzk}_1^{P\to V}, \mathsf{hzk}_2^{V\to P}, \mathsf{hzk}_3^{P\to V})$ and the private randomness of V. Output whatever $\mathsf{Head.ZK}_4$ outputs.
   - If $ch = 1$, for each $i \in SI$, execute the algorithm DWI_4 on $(\mathsf{dwi}_{1,i}^{P\to V}, \mathsf{dwi}_{2,i}^{V\to P}, \mathsf{dwi}_{3,i}^{P\to V})$. If all executions of DWI_4 accept, then output accept and reject otherwise.

Figure 1: 3-round Bounded Rewinding Secure WI

2. **Case 2:** For all $i \in [N]$, c_i is indeed a commitment to 0. Assume that the verifier accepts all $\lambda(\lambda-1)$ executions of the DIWIprotocol. Then w.h.p, the prepared views $V_1, \ldots, V_\lambda$ are such

that each pair (V_i, V_j) is consistent. This follows from the soundness of the DIWI protocol (which has negligible soundness error). Since the underlying MPC construction has perfect correctness, it follows that $x \in L$ which is a contradiction. Hence, w.h.p, the verifier must reject at least one execution of the DIWI protocol.

Suppose the probability of Case 1 and Case 2 are p and $1-p$ respectively. Then RWI has soundness $p\delta/2 + (1-p) \cdot (1 - \mathsf{negl}(\lambda)) \geq \delta/2$.

Witness Indistinguishability under B rewinds: We will now prove that RWI satisfies witness indistinguishability under B rewinds where B is the rewinding parameter of the Head.ZK construction. Consider the following sequence of hybrid experiments.

Hybrid H_0: This hybrid experiment corresponds to the honest protocol execution where the prover uses witnesses $w^1, \ldots, w_0^B$ to prove the statements $x^1, \ldots, x^B$ respectively in B rewound executions.

Hybrid H_1: In this hybrid experiment, the prover starts using the simulator S_{zk} to simulate the execution of the protocol Head.ZK across all executions. In more details, the prover runs S_{zk} to get the message $\mathsf{hzk}_1^{P \to V}$. Prover then prepares the first message of the protocol honestly except for using $\mathsf{hzk}_1^{P \to V}$ given by S_{zk} and sends it to V^*. In all the B execution, the prover handles the messages of Head.ZK as follows. If $ch = 0$, prover forwards the verifier message of Head.ZK to S_{zk} and forwards the response back to V^*. If $ch = 1$, the prover aborts this particular execution with S_{zk} since there will be no further message of Head.ZK in this execution. All messages other than messages of Head.ZK are computed honestly as in H_0.

By the zero-knowledge property of Head.ZK, it follows that the view produced by S_{zk} across the B executions will be indistinguishable from that in H_0. Hence, the view of V^* in H_1 is indistinguishable from that in H_0.

Hybrid H_2: The prover now selects a random set of $\lambda(\lambda-1)$ distinct indices (from N indices) for each of the B executions even before the protocol starts. Denote these sets by $SI_1, \ldots, SI_B$. Define a set SU which consists of all the indices which appear in *more than 1* of these B sets $SI_1, \ldots, SI_B$. In hybrid H_2, the prover is identical to that in H_1 except that for each $i \notin SU$, the prover sets $c_i = \mathsf{Com}(1)$. (The remaining commitments are commitments of 0 as before.)

The indistinguishability of this hybrid follows directly from the hiding property of Com. Observe that in this experiment, the openings of the commitments $c_1, \ldots, c_N$ are not being used by the prover in any of the B executions.

We also prove the following lemma.

Lemma 2. *Suppose $N = B^2\lambda^4$. Except with negligible probability over the random tape of the prover, $|SU| \leq \frac{\lambda}{6}$.*

Proof. Define $T = B\lambda^2$. We consider the following experiment. First pick T independent and random indices from the set N. The (multi)set of indices is denoted by ST and the indices themselves are denoted by $E_1, \ldots, E_T$. Since the indices are picked independently, it is possible that some of them maybe the same (and hence ST is a multiset rather than a set). We now construct sets $SI_1, \ldots, SI_B$ from ST as follows. SI_1 will simply consist of the first $\lambda(\lambda-1)$ mutually distinct elements from ST (starting with element E_1). SI_2 will consist of the second $\lambda(\lambda-1)$ mutually distinct elements from

ST, and so on. Note that for all i, all elements within SI_i must be distinct. However, two sets SI_i and SI_j with $i \neq j$ may have non-zero intersection. To be able to successfully construct $SI_1, \ldots, SI_B$, it is sufficient (though not necessary) for ST to have at least $B\lambda(\lambda - 1)$ distinct elements. The distribution of sets $SI_1, \ldots, SI_B$ constructed using this algorithm is identical to the distribution when $SI_1, \ldots, SI_B$ are picked one at a time by randomly picking $\lambda(\lambda - 1)$ distinct indices out of N. We now prove that, in fact, most elements in ST are distinct.

Claim 3.1. *Multiset ST has at least $T - \lambda/6$ distinct elements except with negligible probability.*

Proof. Since all elements of ST are picked independently and uniformly, the probability that the i-th element is identical to any other element in ST is at most $\frac{T}{N}$. Define random variable X_i s.t. $X_i = 1$ if $\exists j \neq i$ s.t. $E_i = E_j$, and, $X_i = 0$ otherwise. Clearly, the expectation $\mathbb{E}[X_i] \leq \frac{T}{N}$. Denote $X = \sum_i X_i$. By linearly of expectation, $\mathbb{E}[X] \leq \frac{T^2}{N} = 1$.

Denote $\mathbb{E}[X]$ by μ. Set $\delta = \frac{\lambda}{7}$. By Chernoff bounds, we have that $\Pr[X > (1+\delta)\mu] \leq \mathsf{negl}(\lambda)$. Thus, $\Pr[X > \frac{\lambda}{6}] \leq \mathsf{negl}(\lambda)$.

□

If ST has T elements and at least $T - \lambda/6$ are distinct, at most $\lambda/6$ elements appear multiple times in ST. This also means that at most $\lambda/6$ elements appear multiple times across the sets $SI_1, \ldots, SI_B$. Thus, $|SU| \leq \frac{\lambda}{6}$.

□

Hybrid H_3: This hybrid is identical to the previous except in the way prover computes $\{\mathsf{dwi}_{3,i}^{P \to V}\}_{i \notin SU}$ in the last round. Note that if $i \notin SU$, $c_i = \mathsf{Com}(1)$. Hence, the prover now has an alternative witness to prove the statement. The prover switches to using this witness to compute $\{\mathsf{dwi}_{3,i}^{P \to V}\}_{i \notin SU}$ in all executions.

Now observe the following. By definition of SU, if $i \notin SU$, then the message $\{\mathsf{dwi}_{3,i}^{P \to V}\}_{i \notin SU}$ is actually required to be sent in *at most* one execution. That is, $i \notin SU$, the i-th parallel instance of DIWI is only executed at most once (without any rewinding). Hence, the indistinguishability of the view of V^* between H_2 and H_3 follows from the witness indistinguishability of DIWI.

Hybrid H_4: We now define a set $S_{leak} \subset [\lambda]$ of the views as follows. Start with an empty S_{leak}. For all $k_{i,j} \in SU$, add i and j to S_{leak}. Clearly, since $|SU| \leq \frac{\lambda}{6}$, it follows that $|S_{leak}| \leq \frac{\lambda}{3}$.

This hybrid is identical to the previous except now for all $i \notin S_{leak}$, the prover sets cv_i to be $\mathsf{Com}(0)$ as opposed to $\mathsf{Com}(V_i)$ (in all executions). Now observe that if $i \notin S_{leak}$, the opening of cv_i was not being used as a witness in any DIWI execution. This is because any DIWI instance which could have used V_i has already been switched to using the alternate witness. Thus, the indistinguishability of the view of V^* between H_3 and H_4 directly follows from the hiding of the commitment scheme Com.

Hybrid H_5: This hybrid is identical to the previous one except in how the views are computed by the prover in the last round. We note that in each rewound execution, the prover only needs to construct a view V_i if $i \in S_{leak}$. However since $|S_{leak}| \leq \frac{\lambda}{3}$, the prover needs to construct at most $\frac{\lambda}{3}$ views. The prover stops using the supplied witness at this point and instead starts using the MPC simulator to generate all the required views. Observe that we are using an MPC protocol with perfect correct and perfect security which is capable to simulating the view of up to $\frac{\lambda}{3}$ players. Thus, the indistinguishability of the view of V^* between H_4 and H_5 follows from indistinguishability of real and simulated views in the underlying MPC construction.

We now observe that in hybrid H_5, our prover is no longer using the supplied witnesses in any of the B execution. Hence, our construction RWI is, in fact, zero-knowledge under B rewinds. This in particular implies that our construction satisfies the notion of WI with bounded rewind security as defined in 5. We also note that although not necessary in our application, the parallel repetition of RWI can also be shown to have the proof of knowledge property.

3.4 Pairwise-Independent Coin-Flipping

In this section, we combine the building blocks from the two prior sections into a single coin-flipping protocol. This is the only context in which either building block will appear throughout the rest of the paper. At a high level, the coin-flipping protocol consists of the following parts; $n \in \mathbb{N}$ is an integer parameter.

1. an unbounded polynomial commitment (Section 3.1) which C uses to commit to the value N;
2. an $n-$way, three round coin-flip type protocol:
   (a) C sends $\{z_i\}_{i\in[n]}$ where each z_i is a commitment to a random string $\mathbf{r}_i \in \{0,1\}^\lambda$; let $\mathsf{val}(N) = \mathcal{O}(\log N)$ be a parameter we will fix precisely later.
   (b) R sends random strings $\{\mathbf{r}'_i\}_{i\in[n]}$;
   (c) C sends $\{\mathbf{r}_i\}_{i\in[n]}$; we think of the strings $\mathbf{r}_i \oplus \mathbf{r}'_i$ as being the "outputs" of this subroutine;
3. C specifies a family of Goldreich-Levin sets by sending $\{z^{\mathsf{GL}}_\alpha\}_{\alpha\in[\lambda]}$ where each z^{GL}_α is a commitment to a random string $\mathbf{r}^{\mathsf{GL}}_\alpha \in \{0,1\}^\lambda$;
4. C proves using RWI (Section 3.2) that:

   EITHER: $\mathbf{r}_i$ is the committed string inside z_i for all $i \in [n]$;

   OR: there exists $j \in [n]$ such that $\mathbf{r}_j \in \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1, \ldots, \mathbf{r}^{\mathsf{GL}}_{\mathsf{val}})$ where $\mathsf{val} = \mathcal{O}(\log N)$, and $\mathbf{r}_i$ is the committed string inside z_i for all $i \neq j$.

The above is an oversimplification of our actual protocol. For technical reasons resulting from our use of witness-indistinguishable proofs, C actually commits to pairs of strings in Step 3: $\{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\lambda]}$. Then in the 'OR' part of the statement in Step 4, C proves that $\mathbf{r}_j \in \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{b,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{b,\mathsf{val}})$ holds for some $b \in \{0,1\}$. The full protocol appears in Figure 2

3.5 Security Analysis of $\mathsf{PairwiseCF}_n$

Notation. We denote the three round transcript of $\mathsf{PairwiseCF}_n$ by (τ_1, τ_2, τ_3). We say that (τ_1, τ_2, τ_3) is *valid* if the proof verification algorithm accepts $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$.

The following Claim will help us establish extraction properties for our commitment scheme in Section 6.

Claim 1. *For all $\delta > 0$ there exists $M_\delta = \mathsf{poly}\big(\lambda, 1/\delta\big)$ such that for any first message τ_1 of $\mathsf{PairwiseCF}_n$ sent by C^*, there exists a set $\mathsf{LEGAL} \subset \{0,1\}^{\lambda\cdot n}$ of size at most $|\mathsf{LEGAL}| \leq M_\delta$ such that*

$$\Pr\Big[(\tau_1, \tau_2, \tau_3) \text{ valid } \& \ \mathbf{r} \notin \mathsf{LEGAL}\Big] \leq \delta, \tag{2}$$

where the probability is over (τ_2, τ_3).

Parameters and Subroutines: Let λ be the security parameter, and $n = \mathsf{poly}(\lambda)$ a parameter. Let Com be a non-interactive, perfectly binding commitment scheme. Let RWI be a three-round $3-$rewind secure delayed-input WI proof. Let $\mathsf{val}(N) = 2\log(\lambda) + 3\log(N) + 2$.

1. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $\left(c; \{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\lambda]}; \{z_i\}_{i\in[n]}; \mathsf{rwi}_1\right)$ to R where:

(a) $c = \mathrm{Com}(s \circ N; \eta)$ for random $s, N \leftarrow \left[2^\lambda\right]$ and $\eta \leftarrow \$$;

(b) $z^{\mathsf{GL}}_{b,\alpha} = \mathrm{Com}(\mathbf{r}^{\mathsf{GL}}_{b,\alpha}; \omega_{b,\alpha})$ for random $\mathbf{r}^{\mathsf{GL}}_{b,\alpha} \leftarrow \{0,1\}^\lambda$, $\omega_{b,\alpha} \leftarrow \$$;

(c) $z_i = \mathrm{Com}(\mathbf{r}_i; \rho_i)$ for random $\mathbf{r}_i \leftarrow \{0,1\}^\lambda$, $\rho_i \leftarrow \$$;

(d) $\mathsf{rwi}_1 \leftarrow \mathsf{RWI}_1$; the statement will come in the third round;

2. $\underline{\mathrm{R} \longrightarrow \mathrm{C}}$**:** R sends $\left(s'; \{\mathbf{r}'_i\}_{i\in[n]}; \mathsf{rwi}_2\right)$ to C where:

$$\text{(a) } s' \leftarrow \left[2^\lambda\right];\ \text{(b) } \mathbf{r}'_i \leftarrow \{0,1\}^\lambda\ \forall\ i \in [n];\ \text{(c) } \mathsf{rwi}_2 \leftarrow \mathsf{RWI}_2.$$

3. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $\left(\{\mathbf{r}_i\}_{i\in[n]}; \mathsf{rwi}_3\right)$ to R where $\{\mathbf{r}_i\}$ are as in (1c) and $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ prove:

EITHER: $\exists\ \{\rho_i\}_{i\in[n]}$ such that $z_i = \mathrm{Com}(\mathbf{r}_i; \rho_i)\ \forall\ i \in [n]$;

OR: $\exists\ \left(b, j, \mathsf{val}; s, N, \eta; \{\mathbf{r}^{\mathsf{GL}}_\alpha, \omega_\alpha\}_{\alpha\in[\mathsf{val}]}; \{\rho_i\}_{i\neq i^*}\right)$ such that $N > 1$ and

$$\begin{array}{ll} (i)\ b \in \{0,1\};\ j \in [n]; \mathsf{val} = \mathsf{val}(N) & (ii)\ c = \mathrm{Com}(s \circ N; \eta) \\ (iii)\ s + s' \equiv 0 \pmod N & (iv)\ z_i = \mathrm{Com}(\mathbf{r}_i; \rho_i)\ \forall\ i \neq j \\ (v)\ z^{\mathsf{GL}}_{b,\alpha} = \mathrm{Com}(\mathbf{r}^{\mathsf{GL}}_\alpha; \omega_\alpha)\ \forall\ \alpha \in [\mathsf{val}] & (vi)\ \mathbf{r}_j \in \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1, \ldots, \mathbf{r}^{\mathsf{GL}}_{\mathsf{val}}) \end{array}$$

Output: If the proof verification accepts $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$, then both players output $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$.

Figure 2: Pairwise Independent Coin Flipping $-$ $\mathsf{PairwiseCF}_n$

Proof. Let $\tau_1 = \left(c; \{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\lambda]}; \{z_i\}_{i\in[n]}; \mathsf{rwi}_1\right)$. Let $\mathbf{r}^{\mathsf{GL}}_{b,\alpha}$ and $\mathbf{r}^{\mathsf{hon}}_i$ be the committed strings inside $z^{\mathsf{GL}}_{b,\alpha}$ and z_i, respectively. Let N denote the modulus part of the committed string inside c. The soundness of RWI ensures that, except with probability $2^{-\Omega(\lambda)}$, the values $\{\mathbf{r}_i\}$ sent by C^* in the final round must be such that $\mathbf{r}_i = \mathbf{r}^{\mathsf{hon}}_i$ for all i except for possibly $i = i^*$ where it might be that

$$\mathbf{r}_{i^*} \in \mathsf{LEGAL} := \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{0,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{0,\mathsf{val}}) \cup \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{1,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{1,\mathsf{val}})$$

where $\mathsf{val} = 2\log(\lambda) + 3\log(N) + 2$. We assume $N \leq 4/\delta$, since if not, $\mathbf{r}_i = \mathbf{r}^{\mathsf{hon}}_i$ must hold for all i, unless either $s + s' \equiv 0 \pmod N$ (occurs with probability at most $\delta/2$), or if C^* proves a false statement in RWI (occurs with probability $2^{-\Omega(\lambda)}$). Thus, in this case, LEGAL has size 1, consisting only of the strings $\{\mathbf{r}^{\mathsf{hon}}_i\}_{i\in[n]}$. However, if $N \leq 4/\delta$, then $|\mathsf{LEGAL}| \leq 8n\lambda^3N^3 \leq 512n\lambda^3/\delta^3 =: M_\delta$. The Claim follows. $\square$

Simulation Guarantees. We present a simulator which will be critical to the proof of hiding in Section 5, and synchronizing non-malleability in Appendix B.

Lemma 3. *Assume one-to-one one-way functions exist, and let* RWI *be secure against* $B = 2$ *rewinds. There exists a simulator algorithm,* $\mathsf{SIM}_{\mathsf{CF}}$*, satisfying the following syntax, running time, simulation and pairwise independence guarantees.*

- **Syntax:** $\mathsf{SIM}_{\mathsf{CF}}$ *takes as input a triple* $(1^\lambda, 1^n, 1^N)$ *for integer parameters* (λ, n, N) *defining the value* $\mathsf{val} = 2\log(\lambda) + 3\log(N) + 2$*. Additionally,* $\mathsf{SIM}_{\mathsf{CF}}$*, gets oracle access to a (possibly malicious)* R^**, and outputs strings* $\mathbf{r}_\alpha^{\mathsf{GL}} \in \{0,1\}^\lambda$ *for* $\alpha = 1, \ldots, \mathsf{val}$*, one first and two second round messages* $\tau_1, \tau_2, \hat{\tau}_2$*, as well as many third messages* $\{\tau_3(i^*, \overline{\mathbf{r}}), \hat{\tau}_3(i^*, \overline{\mathbf{r}})\}_{i^*, \overline{\mathbf{r}}}$*, where for each* $i^* \in [n]$ *and* $\overline{\mathbf{r}} \in \mathsf{GL}(\mathbf{r}_1^{\mathsf{GL}}, \ldots, \mathbf{r}_{\mathsf{val}}^{\mathsf{GL}})$*,* $\big(\tau_1, \tau_2, \tau_3(i^*, \overline{\mathbf{r}})\big)$ *is a transcript with sub-transcript* $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ *which proves the 'OR' part of the statement using* i^* *and* $\overline{\mathbf{r}}$*; the* $\hat{\tau}_3(i^*, \overline{\mathbf{r}})$ *are analogous.*

- **Running Time:** *The expected running time of* $\mathsf{SIM}_{\mathsf{CF}}$ *is* $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{R}^*}, n, N\big)$.

- **Simulation:** *For all PPT* R^**,* $n, N = \mathsf{poly}(\lambda)$ *and* $j \in [n]$*, the following distributions are computationally indistinguishable:*

  - $\mathsf{REAL}^{\mathrm{R}^*}_{\mathsf{PairwiseCF}_n}$**:** *honest* C *plays* $\mathsf{PairwiseCF}_n$ *twice against* R^**, where* R^* *is allowed to rewind* C *one time, the transcript* $(\tau_1, \tau_2, \tau_3, \hat{\tau}_2, \hat{\tau}_3)$ *is output.*
  - $\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(j)$**:** $\mathsf{SIM}_{\mathsf{CF}}(1^\lambda, 1^n, 1^N)$ *is run with oracle access to* R^**, obtaining* $\{\mathbf{r}_\alpha^{\mathsf{GL}}\}_{\alpha \in [\mathsf{val}]}$*, and* $\big(\tau_1, \tau_2, \{\tau_3(i^*, \overline{\mathbf{r}})\}_{i^*, \overline{\mathbf{r}}}, \hat{\tau}_2, \{\hat{\tau}_3(i^*, \overline{\mathbf{r}})\}_{i^*, \overline{\mathbf{r}}}\big)$*; then a string* $\overline{\mathbf{r}} \leftarrow \mathsf{GL}(\mathbf{r}_1^{\mathsf{GL}}, \ldots, \mathbf{r}_{\mathsf{val}}^{\mathsf{GL}})$ *is chosen and* $\big(\tau_1, \tau_2, \tau_3(j, \overline{\mathbf{r}}), \hat{\tau}_2, \hat{\tau}_3(j, \overline{\mathbf{r}})\big)$ *is output.*

- **Computational Pairwise Independence:** *For non-empty* $S \subset \{1, \ldots, \mathsf{val}\}$ *and* $t \in [\lambda]$*, let* $\Sigma^j_{S,t}$ *denote the output of the process which runs* $\mathsf{SIM}_{\mathsf{CF}}{}^{\mathrm{R}^*}(1^\lambda, 1^n, 1^N)$*, sets* $\overline{\mathbf{r}} = \mathbf{e}_t \oplus \big(\bigoplus_{\alpha \in S} \mathbf{r}_\alpha^{\mathsf{GL}}\big)$ *and outputs* $\big(\tau_1, \tau_2, \tau_3(j, \overline{\mathbf{r}}), \hat{\tau}_2, \hat{\tau}_3(j, \overline{\mathbf{r}})\big)$*. For all PPTs* D *and* $j \in [n]$

$$\Pr_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\left[\exists\, s, t \in [\lambda] \text{ st } \left|\Pr_S\Big[\mathsf{D}\big(\Sigma^j_{S,s}\big) = 1\Big] - \Pr_S\Big[\mathsf{D}\big(\Sigma^j_{S,t}\big) = 1\Big]\right| \geq \frac{1}{N}\right] < \frac{1}{N}. \tag{3}$$

Remark. *Rackoff's proof of the Goldreich-Levin theorem makes crucial use of a concentration bound derived from pairwise independence. The final condition in the Lemma above is a computational version of this concentration bound.*

We prove Lemma 3 in Appendix A. The simulator $\mathsf{SIM}_{\mathsf{CF}}$ is described in Figure 3.

4 Our New Commitment Scheme

In this section we describe our main construction – a new commitment scheme obtained by composing a basic commitment with an interactive version of Blum's non-interactive commitment. The random string used in the second part is jointly computed by C and R using $\mathsf{PairwiseCF}$ from Section 3.4. Specifically, our new commitment scheme consists of the following parts (see Figure 4).

Parameters: Integers $n, N \in \mathbb{N}$, $\mathsf{val} = 2\log(\lambda) + 3\log(N) + 2$.

Continue or Abort: Prepare τ_1, a first message of $\mathsf{PairwiseCF}_n$ honestly and feed τ_1 to R^*; if R^* aborts, output τ_1 and halt. Otherwise, continue.

Main Loop: While true

– **Prepare First Message:** Generate $\tau_1 = \left(c, \{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\lambda]}, \{z_i\}_{i\in[n]}, \mathsf{rwi}_1\right)$ as follows:

· draw $s \leftarrow \left[2^\lambda\right]$, $\eta \leftarrow \$$ set $c = \mathrm{Com}(s \circ N; \eta)$;

· draw $\mathbf{r}^{\mathsf{GL}}_{b,\alpha} \leftarrow \{0,1\}^\lambda$, $\omega_{b,\alpha} \leftarrow \$$ set $z_{b,\alpha} = \mathrm{Com}(\mathbf{r}^{\mathsf{GL}}_{b,\alpha}, \omega_{b,\alpha})$;

· draw $\mathbf{r}_i \leftarrow \{0,1\}^\lambda$, $\rho_i \leftarrow \$$ set $z_i = \mathrm{Com}(\mathbf{r}_i; \rho_i)$;

· draw $\mathsf{rwi}_1 \leftarrow \mathsf{RWI}_1$ (statement will be chosen in third round).

– **Break or Continue:** Feed τ_1 to R^*. If R^* aborts continue. Otherwise, R^* responds with $\tau_2 = \left(s', \{\mathbf{r}'_i\}_{i\in[n]}, \mathsf{rwi}_2\right)$. If $s + s' \not\equiv 0 \pmod N$, continue. Otherwise, break out of the main loop.

Generate the Rewind Thread: Feed R^* once again with τ_1 until R^* responds with another valid second message $\hat\tau_2$ such that $s + \hat s' \equiv 0 \pmod N$.

Output: Output $\left(\{\mathbf{r}^{\mathsf{GL}}_\alpha\}_{\alpha\in[\mathsf{val}]}, \{\mathbf{r}_i\}_{i\in[n]}; \tau_1, \tau_2, \{\tau_3(i^*, \overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}}; \hat\tau_2, \{\hat\tau_3(i^*, \overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}}\right)$,

· the $\mathbf{r}_i$ and $\mathbf{r}^{\mathsf{GL}}_\alpha$ are from the first message, $\mathbf{r}^{\mathsf{GL}}_\alpha = \mathbf{r}^{\mathsf{GL}}_{b,\alpha}$ for a random $b \leftarrow \{0,1\}$.

· for each $i^* \in [n]$ and $\overline{\mathbf{r}} \in \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1, \ldots, \mathbf{r}^{\mathsf{GL}}_{\mathsf{val}})$, $\tau_3(i^*, \overline{\mathbf{r}}) = \left(\{\tilde{\mathbf{r}}_i\}_{i\in[n]}, \mathsf{rwi}_3\right)$, the third message for $\mathsf{PairwiseCF}_n$ where $\tilde{\mathbf{r}}_{i^*} = \overline{\mathbf{r}}$ and $\tilde{\mathbf{r}}_i = \mathbf{r}_i$ for all $i \neq i^*$, and $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ proves the 'OR' part of the statement specified in the protocol. The $\hat\tau_3(i^*, \overline{\mathbf{r}})$ are analogous.

Figure 3: The Simulator $-$ $\mathsf{SIM}_{\mathsf{CF}}$

Building Blocks: Let $\mathrm{Com} = (\mathrm{Com}_1, \mathrm{Com}_2, \mathrm{Com}_3)$ be a commitment scheme with 3 (or fewer) rounds whose decommitment information is fixed after the first round. Let $\mathcal{F}_{\mathsf{1-1\,owf}}$ be a family of one-to-one one-way functions.

Input: C has input, $m \in \{0,1\}^\lambda$, a message to commit to; R has no input.

1. C and R run $\mathrm{Com}(m)$ producing a transcript $(\sigma_1, \sigma_2, \sigma_3)$. Let $c = (c_1, \ldots, c_n) \in \{0,1\}^n$ be the decommitment information.

2. C and R run $\mathsf{PairwiseCF}_n$ (Section 3.4) obtaining transcript (τ_1, τ_2, τ_3) with output $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$.

3. Additionally:

– in the first round C chooses $f_i \leftarrow \mathcal{F}_{1-1\,\mathsf{owf}}$ and $\mathbf{x}_i \leftarrow \{0,1\}^\lambda$ for $i \in [n]$ and sends $\{f_i, \mathbf{y}_i\}_{i\in[n]}$ to R where $\mathbf{y}_i = f_i(\mathbf{x}_i)$;
– in the third round C sends $\{v_i\}_{i\in[n]}$ to R where $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$.

Theorem 4. *Assume that one-to-one one-way functions exist. If* Com *is a* $3-$*round (or fewer) perfectly binding commitment scheme where the decommitment information is fixed after the first message, and* $\mathsf{PairwiseCF}_n$ *is a pairwise independent coinflip protocol, then* $\langle \mathrm{C}, \mathrm{R} \rangle$ *is a three-round, perfectly binding, distributionally extractable commitment scheme that is non-malleable against a sequential MIM.*

We prove Theorem 4 over the next three sections. We prove hiding, distributional extraction, and sequential non-malleability in Sections 5, 6, and 7, respectively. Note that perfect binding follows immediately from the perfect binding of Com and the injectivity of the functions in $\mathcal{F}_{1-1\,\mathsf{owf}}$.

Theorem 5. *Assume that one-to-one one-way functions exist, let* Com *be the main component of the commitment scheme from [GPR16], and let* $\mathsf{PairwiseCF}_n$ *be a pairwise independent coinflipping protocol. Then* $\langle \mathrm{C}, \mathrm{R} \rangle$ *is a three-round, perfectly binding, non-malleable commitment scheme.*

The main component of the GPR scheme is a 3-round perfectly binding commitment scheme whose decommitment only depends on the first message, so Theorem 4 applies. All that is needed for Theorem 5 beyond Theorem 4 is non-malleability against a synchronizing MIM. The main component of GPR is non-malleable against a synchronizing MIM, and moreover the security reduction in the proof rewinds the left execution just one time. Because of the rewind-security of $\mathsf{PairwiseCF}_n$, the reduction from GPR goes through even when the main component is composed with $\mathsf{PairwiseCF}_n$. This observation was already made in GPR. For this reason, the remainder of the main body of this paper is devoted to proving Theorem 4; non-malleability against a synchronizing MIM (*i.e.*, proving Theorem 5) is treated briefly in Appendix B.

5 Hiding

In this section we prove the following lemma.

Lemma 4. *Assume* Com *is a commitment scheme whose decommitment information depends only on the first message,* $\mathcal{F}_{1-1\,\mathsf{owf}}$ *is a family of one-to-one, one-way functions, and* $\mathsf{PairwiseCF}_n$ *a secure pairwise independent coin-flipping protocol. Then* $\langle \mathrm{C}, \mathrm{R} \rangle$ *is computationally hiding.*

Proof. Consider an adversary R^* playing against a challenger $\mathcal{C}$ in the hiding game for $\langle \mathrm{C}, \mathrm{R} \rangle$. Their interaction goes as follows:

1. R^* sends $m_0, m_1 \in \{0,1\}^\lambda$ to $\mathcal{C}$.
2. $\mathcal{C}$ chooses $b \leftarrow \{0,1\}$, then $\mathcal{C}$ and R^* play an execution of $\langle \mathrm{C}, \mathrm{R} \rangle$ where $\mathcal{C}$ commits to m_b. The transcript of this step is $\mathbb{T} = \big(\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3, \{f_i, \mathbf{y}_i, v_i\}_{i\in[n]}\big)$, where $(\sigma_1, \sigma_2, \sigma_3)$ is a transcript of $\mathrm{Com}(m_b)$, (τ_1, τ_2, τ_3) a transcript of $\mathsf{PairwiseCF}_n$ with output $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$ and where $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$ for $\mathbf{x}_i$ such that $f_i(\mathbf{x}_i) = \mathbf{y}_i$ and $(c_1, \ldots, c_n) \in \{0,1\}^n$ the decommitment of $(\sigma_1, \sigma_2, \sigma_3)$.
3. R^* sends $b' \in \{0,1\}$ and wins if $b' = b$.

Subroutines and Parameters: Let $\mathrm{Com} = (\mathrm{Com}_1, \mathrm{Com}_2, \mathrm{Com}_3)$ be a three-round (or fewer) perfectly binding commitment scheme supporting commitments of $\lambda-$bit strings, with $n-$bit decommitments. Moreover, assume that the decommitment information of Com depends only on the first message. Let $\mathsf{PairwiseCF}_n$ be the pairwise independent coin flipping protocol of Section 3.4. Let $\mathcal{F}_{1-1\,\mathsf{owf}}$ be a family of one-to-one, one-way functions.

Input: C has a message $m \in \{0,1\}^\lambda$; R uses no input.

Commit Phase:

1. $\mathrm{C} \longrightarrow \mathrm{R}$**:** C sends $\big(\sigma_1; \{f_i, \mathbf{y}_i\}_{i\in[n]}; \tau_1\big)$ to R, prepared as follows:

(a) $\sigma_1 = \mathrm{Com}_1(m;\omega)$;
(b) $f_i \leftarrow \mathcal{F}_{1-1\,\mathsf{owf}}$ and $\mathbf{y}_i = f_i(\mathbf{x}_i)$ for $\mathbf{x}_i \leftarrow \{0,1\}^\lambda$;
(c) τ_1 is the first round of $\mathsf{PairwiseCF}_n$.

2. $\mathrm{R} \longrightarrow \mathrm{C}$**:** R sends (σ_2, τ_2) to C, second rounds of Com and $\mathsf{PairwiseCF}_n$.

3. $\mathrm{C} \longrightarrow \mathrm{R}$**:** C sends $\big(\sigma_3, \tau_3; \{v_i\}_{i\in[n]}\big)$ to R where:

(a) $\sigma_3 = \mathrm{Com}_3(\sigma_1, \sigma_2; m; \omega)$; let $(c_1, \ldots, c_n) = \mathrm{Decom}(\sigma_1, \sigma_2, \sigma_3; \omega) \in \{0,1\}^n$.
(b) τ_3 is the third round of $\mathsf{PairwiseCF}_n$; let $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$ be the output of (τ_1, τ_2, τ_3);
(c) $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$.

Decommit Phase: C sends $\{\mathbf{x}_i\}_{i\in[n]}$ to R.

Output: R computes $\hat{c}_i = v_i \oplus \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle$, and checks whether $(\hat{c}_1, \ldots, \hat{c}_n)$ is a valid decommitment of $(\sigma_1, \sigma_2, \sigma_3)$. If not, R outputs $\perp$, if so R recovers and outputs m.

Figure 4: A New Three-Round Commitment Scheme $\langle \mathrm{C}, \mathrm{R} \rangle$

Let $\mathsf{G}_0(b)$ denote the above game where $\mathcal{C}$ has chosen the bit b. We show that $\mathsf{G}_0(0) \approx \mathsf{G}_0(1)$ via a hybrid argument. The main steps are:

$\mathsf{G}_1(0)$ same as $\mathsf{G}_0(0)$ except that $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle$;

$\mathsf{G}_1(1)$ same as $\mathsf{G}_1(0)$ except $(\sigma_1, \sigma_2, \sigma_3)$ is a transcript of $\mathrm{Com}(m_1)$;

$\mathsf{G}_0(1)$ same as $\mathsf{G}_1(1)$ except that the v_i are switched back to $\langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$, where $(c_1, \ldots, c_n)$ is the decommitment information for the new $(\sigma_1, \sigma_2, \sigma_3)$.

Suppose for contradiction that a PPT D and non-negligible $\varepsilon > 0$ are such that

$$\big|\mathsf{D}\big(\mathsf{G}_0(0), \mathsf{G}_0(1)\big)\big| := \left|\mathrm{Pr}_{\mathbb{T}\leftarrow\mathsf{G}_0(0)}\Big[\mathsf{D}(\mathbb{T}) = 1\Big] - \mathrm{Pr}_{\mathbb{T}\leftarrow\mathsf{G}_0(1)}\Big[\mathsf{D}(\mathbb{T}) = 1\Big]\right| > \varepsilon.$$

We prove below that $\left|\mathsf{D}\big(\mathsf{G}_0(0),\mathsf{G}_1(0)\big)\right| \leq \varepsilon/3$. Proving $\left|\mathsf{D}\big(\mathsf{G}_0(1),\mathsf{G}_1(1)\big)\right| \leq \varepsilon/3$ is similar. It follows that $\left|\mathsf{D}\big(\mathsf{G}_1(0),\mathsf{G}_1(1)\big)\right| > \varepsilon/3$, which contradicts the hiding of Com. To prove $\left|\mathsf{D}\big(\mathsf{G}_0(0),\mathsf{G}_1(0)\big)\right| \leq \varepsilon/3$, we use $n+1$ sub-hybrids, $\mathsf{H}_0,\ldots,\mathsf{H}_n$. Hybrid H_k is the same as $\mathsf{G}_0(0)$ except that:

$$v_i = \begin{cases} \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i\rangle \oplus c_i, & i > k \\ \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i\rangle, & i \leq k \end{cases}$$

Note that $\mathsf{H}_0 = \mathsf{G}_0(0)$ and $\mathsf{H}_n = \mathsf{G}_1(0)$. We analyze the change from H_{k-1} to H_k. The Lemma follows immediately from the next Claim. □

Claim 2. *Assume the hypotheses of Lemma 4. Then for all $k = 1,\ldots,n$:*

$$\left|\Pr_{\mathbb{T}\leftarrow\mathsf{H}_{k-1}}\Big[\mathsf{D}(\mathbb{T})=1\Big] - \Pr_{\mathbb{T}\leftarrow\mathsf{H}_k}\Big[\mathsf{D}(\mathbb{T})=1\Big]\right| \leq \frac{\varepsilon}{3n}.$$

Proof of Claim 2. Let $N \in \mathbb{N}$ be so that $8/N = \varepsilon/3n$, and let $\delta := 1/N$; this will simplify notations. Assume for contradiction that $\mathsf{D}\big(\mathsf{H}_{k-1},\mathsf{H}_k\big) > 8\delta$. Notice that the only difference between H_{k-1} and H_k is that $v_k = \langle \mathbf{x}_k, \mathbf{r}_k \oplus \mathbf{r}'_k\rangle \oplus c_k$ in H_{k-1} and $v_k = \langle \mathbf{x}_k, \mathbf{r}_k \oplus \mathbf{r}'_k\rangle$ in H_k. If $c_k = 0$ we are done, so we assume $c_k = 1$. Let H'_{k-1} and H'_k be identical to H_{k-1} and H_k except for how $\mathcal{C}$ prepares (τ_1,τ_2,τ_3). In H'_{k-1} and H'_k, $\mathcal{C}$ draws

$$\Big(\{\mathbf{r}^{\mathsf{GL}}_\alpha\}_{\alpha\in[\mathsf{val}]},\tau_1,\tau_2,\{\tau_3(i^*,\overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}},\hat{\tau}_2,\{\hat{\tau}_3(i^*,\overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}}\Big) \leftarrow \mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(1^\lambda,1^n,1^N),$$

$\overline{\mathbf{r}} \leftarrow \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1,\ldots,\mathbf{r}^{\mathsf{GL}}_{\mathsf{val}})$ and sets $\tau_3 = \tau_3(k,\overline{\mathbf{r}})$. By the simulation property of $\mathsf{PairwiseCF}_n$, we have that $\mathsf{D}\big(\mathsf{H}'_{k-1},\mathsf{H}'_k\big) > 8\delta - \mathsf{negl}(\lambda) > 7\delta$. We now describe an inversion algorithm $\mathcal{A}$ which inverts the one-way function, thus arriving at our contradiction.

- **Input:** $\mathcal{A}$ receives $(f,\mathbf{y})$ as input.
- **Random Choices:** $\mathcal{A}$ prepares $\big(\sigma_1,\sigma_2,\sigma_3,\tau_1,\tau_2,\{\tau_3(\overline{\mathbf{r}})\}_{\overline{\mathbf{r}}},\{f_i,\mathbf{y}_i,v_i\}_{i\in[n]}\big)$ as follows:
  - a commitment $(\sigma_1,\sigma_2,\sigma_3) = \mathrm{Com}(m_0)$, with decommitment $(c_1,\ldots,c_n)\in\{0,1\}^n$;
  - $f_1,\ldots,f_n \leftarrow \mathcal{F}_{1-1\,\mathsf{owf}}$, $\mathbf{x}_1,\ldots,\mathbf{x}_n \leftarrow \{0,1\}^\lambda$ and sets $\mathbf{y}_i = f_i(\mathbf{x}_i)$ for all $i\in[n]$ and then replaces $(f_k,\mathbf{y}_k)$ with $(f,\mathbf{y})$;
  - $\big(\{\mathbf{r}^{\mathsf{GL}}_\alpha\}_\alpha;\tau_1,\tau_2,\{\tau_3(i^*,\overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}},\hat{\tau}_2,\{\tau_3(i^*,\overline{\mathbf{r}})\}_{i^*,\overline{\mathbf{r}}}\big) \leftarrow \mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(1^\lambda,1^n,1^N)$, where τ_2, contains $\{\mathbf{r}'_i\}_i$; and all $\tau_3(k,\overline{\mathbf{r}})$ contain $\{\mathbf{r}_i\}_{i\neq k}$; for each $\overline{\mathbf{r}}\in\mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1,\ldots,\mathbf{r}^{\mathsf{GL}}_{\mathsf{val}})$, let $\tau_3(\overline{\mathbf{r}}) = \tau_3(k,\overline{\mathbf{r}})$;
  - $v_i = \langle \mathbf{x}_i, \mathbf{r}_i\oplus\mathbf{r}'_i\rangle \oplus c_i$ for $i>k$; $v_i = \langle \mathbf{x}_i,\mathbf{r}_i\oplus\mathbf{r}'_i\rangle$ for $i<k$;
  - a guess $g'\in\{0,1\}$ for $\langle \mathbf{x}_k,\mathbf{r}'_k\rangle$.
- **The Prediction Subroutine:** $\mathcal{A}$ will utilize the following subroutine Pred:
  - on input $\overline{\mathbf{r}}\in\mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_1,\ldots,\mathbf{r}^{\mathsf{GL}}_{\mathsf{val}})$, Pred chooses a random guess $g\in\{0,1\}$ for $\langle \mathbf{x}_k,\overline{\mathbf{r}}\rangle$ and sets $v_k = g\oplus g'$, thus obtaining a transcript $\mathbb{T}$;
  - if $\mathsf{D}(\mathbb{T})=1$, Pred outputs g, otherwise Pred outputs a random bit.
- **The Inversion Algorithm:** $\mathcal{A}$ works as follows:
  - Choose guesses $b_1,\ldots,b_{\mathsf{val}} \leftarrow \{0,1\}$ for $\langle \mathbf{x}_k,\mathbf{r}^{\mathsf{GL}}_1\rangle,\ldots,\langle \mathbf{x}_k,\mathbf{r}^{\mathsf{GL}}_{\mathsf{val}}\rangle$.

- For $S \subset \{1,\ldots,\mathsf{val}\}$ set $b_S = \bigoplus_{\alpha\in S} b_i$.
- For each non-empty $S \subset \{1,\ldots,\mathsf{val}\}$ and $t \in [\lambda]$, let $\overline{\mathbf{r}}_{S,t} = \mathbf{e}_t \oplus \bigoplus_{\alpha\in S} \mathbf{r}_\alpha^{\mathsf{GL}}$, and let $x_{S,t} = \mathsf{Pred}(\overline{\mathbf{r}}_{S,t}) \oplus b_S$; we think of $x_{S,t}$ as S's guess for the $t-$th bit of $\mathbf{x}$.
- For each $t \in [\lambda]$, let $x_t = \mathsf{MAJORITY}_S\{x_{S,t}\}$.
- If $\mathbf{x} = (x_1,\ldots,x_\lambda) \in \{0,1\}^\lambda$ is such that $f(\mathbf{x}) = \mathbf{y}$, output $\mathbf{x}$, otherwise output $\perp$.

We say that the random choices $\Big(\sigma_1,\sigma_2,\sigma_3,\tau_1,\tau_2,\{\tau_3(\overline{\mathbf{r}})\}_{\overline{\mathbf{r}}},\{f_i,\mathbf{y}_i,v_i\}_{i\in[n]},g'\Big)$ are *good* if the following three events occur:

$\mathbf{E}_1$: $\Pr_{\overline{\mathbf{r}},g}\Big[\mathsf{D}(\mathbb{T})=1\Big|g=\langle\mathbf{x}_k,\overline{\mathbf{r}}\rangle\Big] \geq \Pr_{\overline{\mathbf{r}},g}\Big[\mathsf{D}(\mathbb{T})=1\Big|g\neq\langle\mathbf{x}_k,\overline{\mathbf{r}}\rangle\Big]+4\delta$, where the probabilities are over $\overline{\mathbf{r}} \leftarrow \mathsf{GL}(\mathbf{r}_1^{\mathsf{GL}},\ldots,\mathbf{r}_{\mathsf{val}}^{\mathsf{GL}})$ and $g \leftarrow \{0,1\}$ (*i.e.*, over the randomness of Pred);

$\mathbf{E}_2$: For all $s,t\in[\lambda]$, $|\mathsf{P}_t-\mathsf{P}_s|<\delta$, where $\mathsf{P}_t := \Pr_S\big[\mathsf{Pred}(\overline{\mathbf{r}}_{S,t})=\langle\mathbf{x}_k,\overline{\mathbf{r}}_{S,t}\rangle\big]$, where the probability is over non-empty $S\subset\{1,\ldots,\mathsf{val}\}$.

$\mathbf{E}_3$: $g'=\langle\mathbf{x}_k,\mathbf{r}'_k\rangle$;

Note that $\mathbf{E}_1$ occurs with probability at least 3δ, since $\mathsf{D}\big(\mathsf{H}'_{k-1},\mathsf{H}'_k\big) > 7\delta$. This is because when $g=\langle\mathbf{x}_k,\overline{\mathbf{r}}\rangle$, $\mathbb{T}$ is distributed according to H_k, while if $g\neq\langle\mathbf{x}_k,\overline{\mathbf{r}}\rangle$, $\mathbb{T}$ is distributed according to H_{k-1}. Also, $\mathbf{E}_2$ occurs with probability at least $1-\delta$ by the "pairwise-independence" property of $\mathsf{PairwiseCF}_n$; and $\mathbf{E}_3$ occurs with probability $1/2$ independently of $\mathbf{E}_2$ and $\mathbf{E}_3$. Therefore, the probability that the random choices made by $\mathcal{A}$ are good is at least δ.

We conclude by showing that whenever the choices are good, $\mathcal{A}$ outputs $\mathbf{x}$ with probability at least $2^{-\mathsf{val}}$. To see this, note that if $\mathbf{E}_1$ and $\mathbf{E}_3$ hold, then

$$\Pr_{S,t}\Big[\mathsf{Pred}(\overline{\mathbf{r}}_{S,t})=\langle\mathbf{x}_k,\overline{\mathbf{r}}_{S,t}\rangle\Big] \geq \frac{1}{2}+\delta,$$

where the probability is over non-empty $S\subset\{1,\ldots,\mathsf{val}\}$ and $t\leftarrow[\lambda]$. If $\mathbf{E}_2$ also holds, then $\Pr_S\big[\mathsf{Pred}(\overline{\mathbf{r}}_{S,t})=\langle\mathbf{x}_k,\overline{\mathbf{r}}_{S,t}\rangle\big] > 1/2$ for all $t\in[\lambda]$. Note also that with probability $2^{-\mathsf{val}}$, all guesses $b_1,\ldots,b_{\mathsf{val}}$ made by $\mathcal{A}$ are correct (*i.e.*, equal to $\langle\mathbf{x}_k,\mathbf{r}_1^{\mathsf{GL}}\rangle,\ldots,\langle\mathbf{x}_k,\mathbf{r}_{\mathsf{val}}^{\mathsf{GL}}\rangle$). In this case, $\langle\mathbf{x}_k,\bigoplus_{\alpha\in S}\mathbf{r}_\alpha^{\mathsf{GL}}\rangle = b_S$ for all $S\subset\{1,\ldots,\mathsf{val}\}$, so $\Pr_S\big[x_{S,t}$ is t-th bit of $\mathbf{x}\big] > \frac{1}{2}$ holds for all $t\in[\lambda]$, and $\mathcal{A}$ outputs $\mathbf{x}$. $\square$

6 Distributional Extraction

In this section we prove the following lemma.

Lemma 5. *Assume* Com *is a* $3-$*round perfectly binding commitment scheme with decommitment fixed after the first message. Then for all* $\varepsilon>0$ *there exists an extractor* Ext_ε *satisfying the following syntax, running time and extraction guarantees.*

- **Syntax:** Ext_ε *is parametrized by* $\varepsilon>0$*, gets oracle access to a possibly unbounded cheating* C^**, takes a transcript* $\mathbb{T}$ *of* $\langle\mathrm{C},\mathrm{R}\rangle$ *as input and outputs a message* m *or the symbol* $\perp$.
- **Running Time:** *The running time of* Ext_ε *is* $\mathsf{poly}\big(\lambda,\mathsf{T}_{\mathrm{C}^*},1/\varepsilon\big)$.
- **Extraction:** *For any cheating, unbounded* C^*, $\Delta\big(\mathsf{val}^{\mathrm{C}^*},\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}\big)\leq\varepsilon$*, where* $\mathsf{val}^{\mathrm{C}^*}$ *and* $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}$ *denote the distributions which generate a transcript* $\mathbb{T}$ *by running* $\langle\mathrm{C},\mathrm{R}\rangle$ *between an honest* R *and* C^* *and output* $m=\mathsf{val}(\mathbb{T})$ *(the committed message inside* $\mathbb{T}$*) and* $m=\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$*, respecitvely.*

Notation. The input to our extractor is the transcript $\mathbb{T}$ of an execution of $\langle \mathrm{C}, \mathrm{R} \rangle$ between a cheating committer C^* and honest R. Concretely, $\mathbb{T} = \big(\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3, \{f_i, \mathbf{y}_i, v_i\}_{i \in [n]}\big)$, where:

1. $(\sigma_1, \sigma_2, \sigma_3)$ is a transcript of Com;
2. (τ_1, τ_2, τ_3) is a transcript of $\mathsf{PairwiseCF}_n$ containing the strings $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i \in [n]}$;
3. for each $i \in [n]$, $f_i : \{0,1\}^\lambda \to \{0,1\}^{\mathsf{poly}(\lambda)}$, $\mathbf{y}_i \in \{0,1\}^{\mathsf{poly}(\lambda)}$ and $v_i \in \{0,1\}$.

We say $\mathbb{T}$ is *valid* if (τ_1, τ_2, τ_3) is valid (which recall means that the RWI proof inside (τ_1, τ_2, τ_3) passes verification). We say $\mathbb{T}$ is *correct* if for all $i \in [n]$, $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$ where $(c_1, \ldots, c_n) \in \{0,1\}^n$ is a valid decommitment of $(\sigma_1, \sigma_2, \sigma_3)$, and where $\mathbf{x}_i \in \{0,1\}^\lambda$ is such that $f_i(\mathbf{x}_i) = \mathbf{y}_i$. Note the committed value inside $\mathbb{T}$ is $\perp$ unless $\mathbb{T}$ is valid and correct. Note that whether or not $\mathbb{T}$ is valid can be efficiently checked, whereas whether $\mathbb{T}$ is correct or not cannot. We say that $i \in [n]$ is *correct in* $\mathbb{T}$ (or just that i is *correct* if $\mathbb{T}$ is clear from context) if $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i \rangle \oplus c_i$.

6.1 Extractor Overview

Intuitively, we think of our extractor Ext_ε as having two parts, denoted Ext_1 and Ext_2. The job of Ext_1 is to extract as many of the $\mathbf{x}_i$ as possible. We show that if $\mathbb{T}$ is a valid commitment, then with very high probability, Ext_1 can extract all but at most logarithmically many. The rough idea is the following. First, note that if $\mathbb{T}$ is a valid commitment, then C^*, if rewound and replayed with a new second message, will produce another valid commitment $\hat{\mathbb{T}}$ with non-negligible probability, say $\varepsilon > 0$. In particular, this means that with probability at least ε, every $i \in [n]$ is correct in $\hat{\mathbb{T}}$. Using a Goldreich-Levin-type argument, one can show that if $i \in [n]$ is such that i is correct in $\hat{\mathbb{T}}$ with probability at least $1/2 + \varepsilon$, then $\mathbf{x}_i$ can be recovered by designing a GL-prediction algorithm with advantage ε, and running the corresponding GL-inverter. Thus the main task is to show how to transform the guarantee of "global correctness with probability ε" into a guarantee of "local correctness with probability $1/2 + \varepsilon$." The key point is that for large enough $t = \mathcal{O}(\log \lambda)$,

$$\left(\frac{1}{2} + \varepsilon\right)^t \leq \varepsilon \leq \Pr\Big[i \text{ cor. in } \hat{\mathbb{T}} \; \forall \, i = 1, \ldots, t\Big] = \prod_{i=1}^{t} \Pr\Big[i \text{ cor. in } \hat{\mathbb{T}} \Big| j \text{ cor. in } \hat{\mathbb{T}} \; \forall \, j < i\Big]. \quad (4)$$

It follows that $\Pr\big[i \text{ cor. in } \hat{\mathbb{T}} \big| j \text{ cor. in } \hat{\mathbb{T}} \; \forall \, j < i\big] \geq 1/2 + \varepsilon$ holds for some $i \in [t]$. Thus, any set of t sessions will have at least one session which is correct with large probability *conditioned on other sessions also being correct.* We take advantage of this guarantee by rewinding in such a way so that correctness in those other sessions can be efficiently checked by comparing against $\mathbb{T}$.

By the time Ext_1 finishes, we have $\mathbf{x}_i$ for all but at most logarithmically many, say $\ell = \mathcal{O}(\log \lambda)$, of the $i \in [n]$. Moreover, for the ℓ values of i which remain, we know that C^* plays so that i is correct/incorrect in $\hat{\mathbb{T}}$ with roughly $1/2$ probability each, else we would have recovered these values in Ext_1. We could use the values of $\mathbf{x}_i$ we have already recovered to compute the corresponding c_i, then brute force search through all $n-$bit strings which agree with the recovered c_i in search of the decommitment of $(\sigma_1, \sigma_2, \sigma_3)$. If this decommitment is found, we could recover m and output m with probability $2^{-\ell}$, $\perp$ with probability $1 - 2^{-\ell}$. However, this would not necessarily produce the correct distribution since there still can be correlations among the ℓ outstanding sessions. For example, C^* might flip a coin and if heads, play so that $\hat{\mathbb{T}}$ is fully correct; if tails, play so that every outstanding i is incorrect in $\hat{\mathbb{T}}$. The purpose of Ext_2 is to remove all such correlations from the remaining sessions, by extracting from an outstanding session any time a correlation exists. Procedurally, Ext_2 operates very

much like Ext_1, so we do not explain further here. After Ext_2 is finished (and ℓ the number of remaining sessions is adjusted), the output strategy of outputting m with probability $2^{-\ell}$ and $\perp$ with probability $1 - 2^{-\ell}$ gives the correct distribution.

6.2 Extractor Subroutines

The following claim is proved in Section 6.4.

Claim 3 (Inner Extractor). *For all $\varepsilon > 0$ there exists an extractor $\mathsf{InnerExt}_\varepsilon$ satisfying the following syntax, running time and extraction guarantees.*

- **Syntax:** *$\mathsf{InnerExt}_\varepsilon$ is parametrized by $\varepsilon > 0$, gets oracle access to a possibly unbounded cheating C^*, takes $\xi_1 = \big(\sigma_1, \tau_1, \{f_j, \mathbf{y}_j\}_{j\in[n]}\big)$ and a set*

$$\mathsf{FOUND} = \big\{(j, \mathbf{x}_j) : j \in \mathsf{F} \;\&\; f_j(\mathbf{x}_j) = \mathbf{y}_j\big\} \subset [n] \times \{0,1\}^\lambda$$

  for some $\mathsf{F} \subset [n]$, and outputs either $(i, \mathbf{x}_i) \in [n] \times \{0,1\}^\lambda$ or $\perp$.

- **Running Time:** *The running time of $\mathsf{InnerExt}_\varepsilon$ is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{C}^*}, 1/\varepsilon\big)$.*

- **Extraction:** *Suppose (ξ_1, FOUND) with $|\mathsf{FOUND}| = n - \ell$ are such that either of the following hold:*

  - *$\ell \geq t$ and $\Pr_{\xi_2}\big[\mathbb{T}$ correct and $\mathbb{T}$ valid$\big|\xi_1\big] \geq \varepsilon/3$; or*
  - *$\ell < t$ and $\Pr_{\xi_2}\big[\mathsf{F}$ correct in $\mathbb{T}$ & $\mathbb{T}$ valid$\big|\xi_1\big] \geq \varepsilon/3$ and*

$$\Big|\Pr_{\xi_2}\big[\mathbb{T} \text{ correct}\big|\xi_1 \;\&\; \mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid}\big] - 2^{-\ell}\Big| \geq \frac{\varepsilon}{3}.$$

  Then with probability at least $1 - 2^{-\Omega(\lambda)}$, $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}(\xi_1, \mathsf{FOUND})$ outputs $(i, \mathbf{x}_i)$ such that $i \notin \mathsf{F}$ and $f_i(\mathbf{x}_i) = \mathbf{y}_i$.

The Reconstruction Procedure. The extractors in this and the next section both work by extracting the $\mathbf{x}_i$ one at a time until enough have been obtained to recover the committed message m. The following message recovery procedure, RecoverMsg will be a useful subroutine in the remainder of the paper.

Input: RecoverMsg takes as input a transcript $\mathbb{T} = \big(\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3, \{f_i, \mathbf{y}_i, v_i\}_{i\in[n]}\big)$ and a set $\mathsf{FOUND} = \big\{(i, \mathbf{x}_i) : i \in \mathsf{F} \;\&\; f_i(\mathbf{x}_i) = \mathbf{y}_i\big\} \subset [n] \times \{0,1\}^\lambda$, where $\mathsf{F} \subset [n]$ has size $|\mathsf{F}| = n - \ell$.

– For each $i \in \mathsf{F}$, let $c_i = v_i \oplus \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}'_i\rangle$, where the strings $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$ are from (τ_1, τ_2, τ_3). Brute force search through the set $\big\{\mathbf{s} \in \{0,1\}^n : s_i = c_i \;\forall\, i \in \mathsf{F}\big\}$ (this set has size 2^ℓ), in search of the decommitment information of $(\sigma_1, \sigma_2, \sigma_3)$. If the decommitment information is not found, output $\perp$ and halt, otherwise let m be the committed message inside $(\sigma_1, \sigma_2, \sigma_3)$.

Output: Output m with probability $2^{-\ell}$, output $\perp$ with probability $1 - 2^{-\ell}$.

6.3 Extractor Construction

Let $\varepsilon > 0$ be a parameter and set $t \in \mathbb{N}$ so that $(2/3)^t = \varepsilon/3$. Let $\mathsf{InnerExt}_\varepsilon$ be the inner extractor of Claim 3, and let RecoverMsg be the message recovery subroutine.

- **Input and Parameters:** Ext_ε takes a transcript $\mathbb{T} = (\xi_1, \xi_2, \xi_3)$ as input and gets oracle access to C^*.
- **Initialize:** If $\mathbb{T}$ is not valid, Ext_ε outputs $\perp$ and halts. Otherwise, a set $\mathsf{FOUND} \subset [n] \times \{0,1\}^\lambda$ is initialized to $\emptyset$. Let $\mathsf{F} \subset [n]$ be the set of i which appear as the first coordinate of an element of FOUND; so F is also initialized to $\emptyset$.
- **Main Loop:** While $|\mathsf{FOUND}| < n$ do the following:
  - run $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}(\xi_1, \mathsf{FOUND})$; if $(i, \mathbf{x}_i)$ is output, update $\mathsf{FOUND} = \mathsf{FOUND} \cup \{(i, \mathbf{x}_i)\}$ and $\mathsf{F} = \mathsf{F} \cup \{i\}$ and continue; if $\perp$ is output, break out of the loop.
- **Output:** If $|\mathsf{FOUND}| \geq n - t$, Ext_ε outputs $\mathsf{RecoverMsg}(\mathbb{T}, \mathsf{FOUND})$; otherwise $\perp$.

Proof of Lemma 5 Assuming Claim 3. It is clear that Ext_ε has the required syntax. The running time of the main loop is dominated by $n = \mathsf{poly}(\lambda)$ times the running time of $\mathsf{InnerExt}_\varepsilon$, which by Claim 3 is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{C}^*}, 1/\varepsilon\big)$. The running time of RecoverMsg is $\mathsf{poly}(\lambda, 2^t) = \mathsf{poly}(\lambda)$. Therefore, the total runtime is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{C}^*}, 1/\varepsilon\big)$, and so it remains to show that $\Delta\big(\mathsf{val}^{\mathrm{C}^*}, \mathsf{Ext}_\varepsilon^{\mathrm{C}^*}\big) \leq \varepsilon$.

Note that if $\mathbb{T}$ is not valid, then $\mathsf{val}(\mathbb{T}) = \perp$, and $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ outputs $\perp$ during initialization. Therefore, the statistical distance in this case is 0 and we can assume that $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ enters the main loop. There are three possible reasons for $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ to exit the main loop:

1. $|\mathsf{FOUND}| = n$;
2. $|\mathsf{FOUND}| < n - t$ and $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}(\xi_1, \mathsf{FOUND}) = \perp$;
3. $|\mathsf{FOUND}| \geq n - t$ and $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}(\xi_1, \mathsf{FOUND}) = \perp$.

In case 1, $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ outputs $\mathsf{val}(\mathbb{T})$ with probability 1, and so the statistical distance is 0 in this case. In case 2, $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ outputs $\perp$ and so $\mathsf{p}_2 := \Pr_{\mathbb{T}}\big[\text{Case 2 occurs } \& \ \mathsf{val}(\mathbb{T}) \neq \perp\big]$ is the statistical distance in this case. Let us say that ξ_1 is *good* if $\Pr_{\xi_2}\big[\mathbb{T} \text{ correct } \& \ \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \varepsilon/3$. We have

$$\mathsf{p}_2 \leq \Pr_{\mathbb{T}}\big[\mathsf{val}(\mathbb{T}) \neq \perp \ \& \ \xi_1 \text{ not good}\big] + \Pr_{\mathbb{T}}\big[\text{Case 2 occurs} \big| \xi_1 \text{ good}\big] < \varepsilon/3 + 2^{-\Omega(\lambda)},$$

using the first extraction guarantee of $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}$ in Claim 3. Finally, consider case 3. In this case $\mathsf{Ext}_\varepsilon^{\mathrm{C}^*}(\mathbb{T})$ outputs $\perp$ if F is not correct in $\mathbb{T}$ and otherwise outputs m with probability $2^{-\ell}$, $\perp$ with probability $1 - 2^{-\ell}$, where m is the committed message inside $(\sigma_1, \sigma_2, \sigma_3)$. If F is not correct in $\mathbb{T}$ then $\mathsf{val}(\mathbb{T}) = \perp$ and the statistical distance is 0. Now, we identify two types of good ξ_1:

1. say $\xi_1 \in \mathsf{G}_1$ if $\Pr_{\xi_2}\big[\mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \varepsilon/3$;
2. say $\xi_1 \in \mathsf{G}_2$ if $\Big|\Pr_{\xi_2}\big[\mathbb{T} \text{ correct} \big| \xi_1 \ \& \ \mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid}\big] - 2^{-\ell}\Big| \geq \varepsilon/3$.

By the second extraction guarantee of Claim 3, $\Pr\big[\text{Case 3 occurs}\big|\xi_1 \in \mathsf{G}_1 \cap \mathsf{G}_2\big] = 2^{-\Omega(\lambda)}$, so we must bound the statistical distance in Case 3 when F is correct in $\mathbb{T}$ and when $\xi_1 \notin \mathsf{G}_1 \cap \mathsf{G}_2$. Let $\mathbf{E}$ be shorthand for the event "F is correct in $\mathbb{T}$ and $\mathbb{T}$ is valid". The statistical distance in Case 3 when $\mathbf{E}$ occurs is

$$\Pr_{\mathbb{T}}\big[\mathbf{E}\big] \cdot \Big|\Pr_{\mathbb{T}}\big[\mathsf{val}(\mathbb{T}) \neq \perp\big|\mathbf{E}\big] - \Pr_{\mathbb{T}}\big[\mathsf{Ext}_{\varepsilon}^{\mathrm{C}^*}(\mathbb{T}) \neq \perp\big|\mathbf{E}\big]\Big| = \Pr_{\mathbb{T}}\big[\mathbf{E}\big] \cdot \Big|\Pr_{\mathbb{T}}\big[\mathbb{T} \text{ correct}\big|\mathbf{E}\big] - 2^{-\ell}\Big|,$$

since when $\mathbf{E}$ occurs, the only possibile values for $\mathsf{val}(\mathbb{T})$ are $\perp$ and the decommitment of $(\sigma_1, \sigma_2, \sigma_3)$. However, this means the statistical distance we want is upper bounded by $\varepsilon/3$, since when $\xi_1 \notin \mathsf{G}_1$, $\Pr_{\mathbb{T}}\big[\mathbf{E}\big] < \varepsilon/3$, and when $\xi_1 \notin \mathsf{G}_2$, $\big|\Pr_{\mathbb{T}}\big[\mathbb{T} \text{ correct}\big|\mathbf{E}\big] - 2^{-\ell}\big| < \varepsilon/3$. Putting everything together, $\Delta\big(\mathsf{val}^{\mathrm{C}^*}, \mathsf{Ext}_{\varepsilon}^{\mathrm{C}^*}\big) \leq \varepsilon/3 + \varepsilon/3 + 2^{-\Omega(\lambda)} < \varepsilon$. □

6.4 Inner Extractor Construction

- **Input and Parameters:** $\mathsf{InnerExt}_{\varepsilon}$ is parameterized by $\varepsilon > 0$. Let $t \in \mathbb{N}$ be such that $\big(2/3\big)^t = \varepsilon/3$, and let $N = \mathsf{poly}(\lambda) \in \mathbb{N}$ and non-negligible $\eta = \eta(\lambda) > 0$ be parameters which we will fix later. $\mathsf{InnerExt}_{\varepsilon}$ takes (ξ_1, FOUND) as input where $\xi_1 = \big(\sigma_1, \tau_1, \{f_j, \mathbf{y}_j\}_{j\in[n]}\big)$ and

$$\mathsf{FOUND} = \big\{(j, \mathbf{x}_j) : j \in \mathsf{F} \;\&\; f_j(\mathbf{x}_j) = \mathbf{y}_j\big\} \subset [n] \times \{0,1\}^{\lambda},$$

  for a set $\mathsf{F} \subset [n]$ of size $n - \ell$. $\mathsf{InnerExt}_{\varepsilon}$ gets oracle access to C^*.

- **Construct $\mathcal{I}$:** If $\ell > t$, let $T \subset [n] \setminus \mathsf{F}$ be a set of size t; if $\ell \leq t$, let $T = [n] \setminus \mathsf{F}$. Let

$$\mathcal{I} = \big\{(i, S) : i \in T \;\&\; S \subset T \setminus \{i\}\big\}.$$

  Note $|\mathcal{I}| \leq t \cdot 2^t$ holds regardless of whether $\ell > t$ or $\ell \leq t$.

- **Outer Loop:** For each $(i, S) \in \mathcal{I}$:

  - **Inner Loop:** Do the following N times, or until $\mathbf{x}_i$ is recovered.

    · **Set the Main Thread:** Choose $\xi_2 = \big(\sigma_2, \tau_2\big)$ honestly where τ_2 contains $\{\mathbf{r}'_j\}_{j\in[n]}$, send ξ_2 to C^* and receive $\xi_3 = \big(\sigma_3, \tau_3, \{v_j\}_j\big)$, where τ_3 contains $\{\mathbf{r}_j\}_j$. Let $\mathbb{T} = (\xi_1, \xi_2, \xi_3)$; if $\mathbb{T}$ is not valid, continue.

    · **Define the GL-Predictor:** Let Pred be the GL-predictor which, on input $\mathbf{s} \in \{0,1\}^{\lambda}$:

      (a) rewinds C^* and sends $\hat{\xi}_2 = (\sigma_2, \hat{\tau}_2)$ where the only difference between $\hat{\tau}_2$ and τ_2 is the distribution of the strings $\{\hat{\mathbf{r}}'_j\}_{j\in[n]}$:

$$\hat{\mathbf{r}}'_j = \begin{cases} \mathbf{s}, & j = i \\ \mathbf{r}'_j, & j \in S \\ \text{random} \leftarrow \{0,1\}^{\lambda}, & j \notin S \cup \{i\} \end{cases}$$

      (b) receive $\hat{\xi}_3 = (\hat{\sigma}_3, \hat{\tau}_3, \{\hat{v}_j\}_{j\in[n]})$ from C^* where $\hat{\tau}_3$ contains $\{\hat{\mathbf{r}}_j\}_j$, and output $\hat{v}_i$ if the following checks pass, otherwise output a random bit. The conditions checked are: 1) $\hat{\mathbb{T}} = (\xi_1, \hat{\xi}_2, \hat{\xi}_3)$ is valid; 2) $\hat{\mathbf{r}}_j = \mathbf{r}_j$ for all $j \in [n]$; 3) $\hat{v}_j = v_j$ for all $j \in S$; 4) $\hat{v}_j \oplus v_j = \langle \mathbf{x}_j, \mathbf{r}'_j \oplus \hat{\mathbf{r}}'_j \rangle$ for all $j \in \mathsf{F}$.

    · **Run the GL-Inverter and Give Output:** Run the GL-inverter Inv corresponding to Pred and advantage η. If Inv recovers $\mathbf{x}_i \in \{0,1\}^{\lambda}$ such that $f_i(\mathbf{x}_i) = \mathbf{y}_i$, output $(i, \mathbf{x}_i)$; otherwise continue.

- **Inner Loop Fail:** This instruction is reached if the inner loop has been executed N times and $(i, \mathbf{x}_i)$ has not been output; in this case continue to the next element of $\mathcal{I}$.

- **Outer Loop Fail:** This instruction is reached if $\mathsf{InnerExt}_\varepsilon$ fails to output $(i, \mathbf{x}_i)$ for any of the $(i, S) \in \mathcal{I}$; in this case, output $\perp$.

Proof of Claim 3. It is clear that $\mathsf{InnerExt}_\varepsilon$ satisfies the syntax requirements, and that the running time is $\mathsf{poly}\big(\lambda, 2^t, N, \mathsf{T}_{\mathrm{C}^*}, 1/\eta\big) = \mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{C}^*}, 1/\varepsilon\big)$. We establish the extraction guarantee. Fix an $\varepsilon > 0$ and cheating, unbounded C^*. Let us say that the input to $\mathsf{InnerExt}_\varepsilon$, (ξ_1, FOUND) with $|\mathsf{FOUND}| = n - \ell$, is *good* if either of the following hold:

- $\ell > t$ and $\Pr_{\xi_2}\big[\mathbb{T} \text{ correct and valid} \big| \xi_1\big] \geq \varepsilon/3$; or
- $\ell \leq t$ and $\Pr_{\xi_2}\big[\mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \varepsilon/3$ and

$$\Big|\Pr_{\xi_2}\big[\mathbb{T} \text{ correct} \big| \xi_1 \;\&\; \mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid}\big] - 2^{-\ell}\Big| \geq \frac{\varepsilon}{3}.$$

We must show that whenever (ξ_1, FOUND) is good, $\mathsf{InnerExt}_\varepsilon^{\mathrm{C}^*}(\xi_1, \mathsf{FOUND})$ outputs a one-way preimage $(i, \mathbf{x}_i)$ with high probability $1 - 2^{-\Omega(\lambda)}$. Towards this end, we define notions of *good* for the intermediate values which arise during the execution of $\mathsf{InnerExt}_\varepsilon$. The intuition is that good inputs give rise to good intermediate values which enable successful extraction. To begin, let $\delta = (\varepsilon/3) \cdot 2^{-t}$, and say that $(i, S) \in \mathcal{I}$ is *good* if both of the following hold:

1. $\Pr_{\xi_2}\big[S \cup \mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \delta$; and
2. $\Big|\Pr_{\xi_2}\big[i \text{ correct in } \mathbb{T} \big| S \cup \mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \;\&\; \xi_1\big] - \frac{1}{2}\Big| \geq \delta$.

Claim 4. *If (ξ_1, FOUND) is good then some $(i, S) \in \mathcal{I}$ is good.*

We prove Claim 4 below, outside of the current proof. For now, we finish the proof of Claim 3 by showing that when (i, S) is good, one of the executions of the inner loop will recover $\mathbf{x}_i$ with high probability. For this purpose, we let $\eta = \delta^{10}$, and we say that a main thread $\mathbb{T} = (\xi_1, \xi_2, \xi_3)$ (chosen during the first step of an inner loop execution) containing the strings $\{\mathbf{r}_j, \mathbf{r}'_j\}_{j \in [n]}$ is *good* if the following all hold:

1. $\mathsf{F} \cup S$ is correct in $\mathbb{T}$ and $\mathbb{T}$ is valid;
2. $\Pr_{\hat{\xi}_2}\big[S \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r} \in \hat{\xi}_3 \big| \mathbf{r}'_S \in \hat{\xi}_2\big] \geq \eta$; and
3. $\Big|\Pr_{\hat{\xi}_2}\big[i \text{ correct in } \hat{\mathbb{T}} \big| S \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r}'_S \in \hat{\xi}_2 \;\&\; \mathbf{r} \in \hat{\xi}_3\big] - \frac{1}{2}\Big| \geq \eta$;

where the probabilities are over $\hat{\xi}_2 = (\sigma_2, \hat{\tau}_2)$ where $\hat{\tau}_2$ is drawn randomly such that it contains $\{\mathbf{r}'_j\}_{j \in S}$ (the events $\mathbf{r}'_S \in \hat{\xi}_2$ and $\mathbf{r} \in \hat{\xi}_3$ indicate that $\hat{\tau}_2$ and $\hat{\tau}_3$ contain the strings $\{\mathbf{r}'_j\}_{j \in S}$ and $\{\mathbf{r}_j\}_{j \in [n]}$).

Claim 5. *If $(i, S) \in \mathcal{I}$ is good, then with probability $1 - 2^{-\Omega(\lambda)}$, at least one of the ξ_2's drawn during the inner loops is good.*

We prove Claim 5 below, also outside of the current proof. We complete the proof of Claim 3 by observing that whenever ξ_2 is good, the GL-prediction function Pred has advantage η^2. By Lemma 1, this means that the GL-inversion algorithm Inv recovers $\mathbf{x}_i$ with probability $1 - 2^{-\Omega(\lambda)}$.

Recall Pred is parametrized with a transcript $\mathbb{T}$ which includes the quantities $\{\mathbf{r}_j, \mathbf{r}'_j, v_j\}_{j\in[n]}$, and rewinds C^* obtaining a new transcript with quantities $\{\hat{\mathbf{r}}_j, \hat{\mathbf{r}}'_j, \hat{v}_j\}_{j\in[n]}$. Then Pred checks whether the four conditions hold: 1) $\hat{\mathbb{T}}$ valid; 2) $\hat{\mathbf{r}}_j = \mathbf{r}_j$ for all $j \in [n]$; 3) $\hat{v}_j = v_j$ for all $j \in S$; 4) $v_j \oplus \hat{v}_j = \langle \mathbf{x}_j, \mathbf{r}'_j \oplus \hat{\mathbf{r}}'_j \rangle$ for all $j \in \mathsf{F}$. If all checks pass, Pred outputs $\hat{v}_i$, otherwise Pred outputs a random bit. We show that if $\mathbb{T}$ is good, then:

1. $\Pr_{\hat{\xi}_2}\big[\text{all checks pass} \big| \mathbf{r}'_S \in \hat{\xi}_2\big] \geq \eta$; and
2. $\Big|\Pr_{\hat{\xi}_2}\big[\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}_i \rangle \big| \text{all checks pass } \& \ \mathbf{r}'_S \in \hat{\xi}_2\big] - \frac{1}{2}\Big| \geq \eta.$

Together these imply Pred has advantage η^2, and the result follows. By definition, since $\mathbb{T}$ is good, $\mathsf{F} \cup S$ is correct in $\mathbb{T}$. Therefore, $j \in [n]$ is correct in $\hat{\mathbb{T}}$ if and only if $v_j \oplus \hat{v}_j = \langle \mathbf{x}_j, \mathbf{r}_j \oplus \hat{\mathbf{r}}_j \oplus \mathbf{r}'_j \oplus \hat{\mathbf{r}}'_j \rangle$. If $\mathbf{r}_j = \hat{\mathbf{r}}_j$ for all $j \in [n]$, then the above condition simplies to $v_j \oplus \hat{v}_j = \langle \mathbf{x}_j, \mathbf{r}'_j \oplus \hat{\mathbf{r}}'_j \rangle$. If $j \in S$, then $\mathbf{r}'_j = \hat{\mathbf{r}}'_j$ in which case the condition simplifies even further to $v_j = \hat{v}_j$. Thus, whenever $\mathsf{F} \cup S$ is correct in $\hat{\mathbb{T}}$, conditions (3) and (4) will pass, and so point 1 above follows from the second point in the definition of good $\mathbb{T}$. Similarly, point 2 above follows from the third point of the definition since when i is correct in $\hat{\mathbb{T}}$, $\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}'_i \rangle \oplus \big(v_i \oplus \langle \mathbf{x}_i, \mathbf{r}'_i \rangle\big)$. Note, the term in parentheses depends only on $\mathbb{T}$ and so does not affect the absolute value of Pred's advantage. □

6.5 Proofs of the Supporting Claims

Proof of Claim 4. Let $\delta = (\varepsilon/3) \cdot 2^{-t}$. Suppose (ξ_1, FOUND) is good with $|\mathsf{FOUND}| = n - \ell$. If $\ell \geq t$, then $\delta \leq \varepsilon/3 \leq \Pr\big[\mathbb{T} \text{ correct } \& \ \mathbb{T} \text{ valid} \big| \xi_1\big] \leq \Pr\big[S \cup \mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid} \big| \xi_1\big]$ holds for all $S \subset [n]$. Moreover, if $T = \{i_1, \ldots, i_t\}$, then

$$\begin{aligned}\left(\frac{1}{2} + \delta\right)^t &\leq \Pr_{\xi_2}\Big[\{i_1, \ldots, i_t\} \text{ correct in } \mathbb{T} \Big| \mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid } \& \ \xi_1\Big] \\ &= \prod_{\alpha=1}^{t} \Pr_{\xi_2}\Big[i_\alpha \text{ correct in } \mathbb{T} \Big| \mathsf{F} \cup \{i_\beta : \beta < \alpha\} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid } \& \ \xi_1\Big],\end{aligned}$$

since $\big(1/2 + \delta\big)^t \leq \varepsilon/3$. Therefore, some $\alpha \in [t]$ is such that

$$\Pr_{\xi_2}\Big[i_\alpha \text{ correct in } \mathbb{T} \Big| \mathsf{F} \cup \{i_\beta : \beta < \alpha\} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid } \& \ \xi_1\Big] \geq \frac{1}{2} + \delta.$$

Therefore, (i, S) is good where $i = i_\alpha$ and $S = \{i_\beta : \beta < \alpha\}$.

The situation is similar when $\ell < t$. In this case, $\Pr_{\xi_2}\big[\mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \varepsilon/3$ and also $\Big|\Pr_{\xi_2}\big[T \text{ correct} \big| \mathsf{F} \text{ correct } \& \ \mathbb{T} \text{ valid } \& \ \xi_1\big] - 2^{-\ell}\Big| \geq \varepsilon/3$. If the quantity inside the absolute value is positive then $\Pr_{\xi_2}\big[\mathsf{F} \cup S \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \frac{\varepsilon}{3} \cdot \big(2^{-\ell} + \frac{\varepsilon}{3}\big) \geq \delta$ holds for all $S \subset T$. Moreover,

$$\begin{aligned}\left(\frac{1}{2} + \delta\right)^\ell &\leq \Pr_{\xi_2}\Big[\{i_1, \ldots, i_\ell\} \text{ correct in } \mathbb{T} \Big| \mathsf{F} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid } \& \ \xi_1\Big] \\ &= \prod_{\alpha=1}^{\ell} \Pr_{\xi_2}\Big[i_\alpha \text{ correct in } \mathbb{T} \Big| \mathsf{F} \cup \{i_\beta : \beta < \alpha\} \text{ correct in } \mathbb{T} \ \& \ \mathbb{T} \text{ valid } \& \ \xi_1\Big],\end{aligned}$$

where $T = \{i_1, \ldots, i_\ell\}$. We have used $\big(1/2 + \delta\big)^\ell \leq 2^{-\ell} + \varepsilon/3$. As above, this implies there exists $(i, S) \in \mathcal{I}$ such that $\Pr_{\xi_2}\big[i \text{ correct in } \mathbb{T} \big| \mathsf{F} \cup S \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \;\&\; \xi_1\big] \geq 1/2 + \delta$.

Finally, when $\ell < t$ and the quantity inside the absolute value is negative,

$$\begin{aligned}\left(\frac{1}{2} - \delta\right)^\ell &\geq \Pr_{\xi_2}\big[T \text{ correct in } \mathbb{T} \big| \mathsf{F} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \;\&\; \xi_1\big] \\ &= \prod_{\alpha=1}^{\ell} \Pr_{\xi_1}\big[i_\alpha \text{ correct in } \mathbb{T} \big| \mathsf{F} \cup \{i_\beta : \beta < \alpha\} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \;\&\; \xi_1\big],\end{aligned}$$

since $\big(1/2 - \delta\big)^\ell \geq 2^{-\ell} - \varepsilon/3$. Let α be minimal such that

$$\Pr_{\xi_2}\big[i_\alpha \text{ correct in } \mathbb{T} \big| \mathsf{F} \cup \{i_\beta : \beta < \alpha\} \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \;\&\; \xi_1\big] \leq 1/2 - \delta,$$

and let $i = i_\alpha$, $S = \{i_\beta : \beta < \alpha\}$. Then $\Pr_{\xi_2}\big[\mathsf{F} \cup S \text{ correct in } \mathbb{T} \;\&\; \mathbb{T} \text{ valid} \big| \xi_1\big] \geq \frac{\varepsilon}{3} \cdot \big(1/2 - \delta\big)^\ell \geq \delta$. Claim 4 follows. □

Proof of Claim 5. Let $\eta = \delta^{10}$ and suppose (i, S) is good; recall there is a pair (ξ_1, FOUND) underlying the good pair (i, S). This means that a second message ξ_2, when fed to C^* specifies an entire transcript $\mathbb{T} = (\xi_1, \xi_2, \xi_3)$. Recall when ξ_2 and $\hat{\xi}_2$ are both chosen, specifying transcripts $\mathbb{T}$ and $\hat{\mathbb{T}}$, we write $\mathbf{r}'_S \in \hat{\xi}_2$ and $\mathbf{r} \in \hat{\xi}_3$ to indicate that the strings $\{\mathbf{r}'_j\}_{j \in S}$ and $\{\mathbf{r}_j\}_{j \in [n]}$ from ξ_2 and ξ_3 appear in $\hat{\xi}_2$ and $\hat{\xi}_3$. Recall, ξ_2 is good if the following three conditions hold:

1. $\mathsf{F} \cup S$ is correct in $\mathbb{T}$ and $\mathbb{T}$ is valid;
2. $\Pr_{\hat{\xi}_2}\big[S \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r} \in \hat{\xi}_3 \big| \mathbf{r}'_S \in \hat{\xi}_2\big] \geq \eta$; and
3. $\left|\Pr_{\hat{\xi}_2}\big[i \text{ correct in } \hat{\mathbb{T}} \big| S \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r}'_S \in \hat{\xi}_2 \;\&\; \mathbf{r} \in \hat{\xi}_3\big] - \frac{1}{2}\right| \geq \eta$.

Given $\mathbf{r} \in \{0,1\}^{\lambda \cdot n}$ and $\mathbf{r}'_S \in \{0,1\}^{\lambda \cdot |S|}$, we define the values

- $\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S) := \Pr_{\hat{\xi}_2}\Big[S \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r} \in \hat{\xi}_3 \Big| \mathbf{r}'_S \in \hat{\xi}_2\Big]$;
- $\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) := \Pr_{\hat{\xi}_2}\Big[S \cup \{i\} \cup \mathsf{F} \text{ correct in } \hat{\mathbb{T}} \;\&\; \hat{\mathbb{T}} \text{ valid} \;\&\; \mathbf{r} \in \hat{\xi}_3 \Big| \mathbf{r}'_S \in \hat{\xi}_2\Big]$.

We think of $\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)$ and $\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S)$ as random variables over $\mathbf{r}'_S \leftarrow \{0,1\}^{\lambda \cdot |S|}$ for $\mathbf{r}$ fixed. We show that whenever (i, S) is good, there exists $\mathbf{r} \in \{0,1\}^{\lambda \cdot n}$ such that the quantity

$$\mathsf{p}_{\mathbf{r}} := \Pr_{\mathbf{r}'_S \leftarrow \{0,1\}^{\lambda \cdot |S|}}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S) \geq \eta \;\&\; \left|\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)}{2}\right| \geq \eta \cdot \mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right]$$

is at least $\delta^8/(2^{12} n \lambda^3)$. Claim 5 follows from this observation. To see this, note that if ξ_2 is such that: (a) $S \cup \mathsf{F}$ is correct in $\mathbb{T}$ and $\mathbb{T}$ is valid; and (b) $\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S) \geq \eta$ and $\left|\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{1}{2} \cdot \mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right| \geq \eta \cdot \mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)$ where $\mathbf{r}$ is the value from $\mathbb{T}$; then ξ_2 is good. The observation says that with probability at least $\delta^8/(2^{12} n \lambda^3)$, there is some $\mathbf{r}$ for which (b) holds. By definition of $\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)$, whenever (b) holds, the probability that (a) holds and $\mathbf{r} \in \xi_3$ is at least η. Therefore, the probability that ξ_2 is good is at least $\eta\delta^8/(2^{12} n \lambda^3)$, and so the expected number of good ξ_2 which appear during the N executions of the inner loop is at least $N\eta\delta^8/(2^{12} n \lambda^3)$. So choose $N \geq (2^{12} n \lambda^4)/\delta^{18}$, so that the expected number of good ξ_2 is at least λ. Then by the Chernoff-Hoeffding inequality, the probability that no good ξ_2 occur during the loop is at most $1 - 2^{-\Omega(\lambda)}$. This completes the proof of Claim 5, and so it remains to lower bound $\max_{\mathbf{r}}\{\mathsf{p}_{\mathbf{r}}\}$ when (i, S) is good.

By definition, if (i, S) is good then the following both hold:

- $\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right] \geq \delta$; and
- $\left|\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S)\right] - \frac{1}{2}\cdot\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right]\right| \geq \delta \cdot \sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right]$.

Let $\mathbb{E} := \sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\left|\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{1}{2}\cdot\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right| - \eta\cdot\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right]$ be shorthand. We have

$$\mathbb{E} \geq \left|\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)}{2}\right]\right| - \delta\cdot\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right] + (\delta-\eta)\cdot\sum_{\mathbf{r}} \mathbb{E}_{\mathbf{r}'_S}\left[\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right] \geq (\delta-\eta)\cdot\delta.$$

On the other hand,

$$\mathbb{E} \leq \frac{\delta^2}{2} + M\cdot\max_{\mathbf{r}}\left\{\mathbb{E}_{\mathbf{r}'_S}\left[\left|\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)}{2}\right| - \eta\cdot\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right]\right\},$$

follows from Claim 1, where $M := M_{\delta^2/2}$. Moreover, as $0 \leq \mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) \leq \mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)$ holds for all $(\mathbf{r}, \mathbf{r}'_S)$, $\mathbb{E}_{\mathbf{r}'_S}\left[\left|\mathsf{Y}_{\mathbf{r}}(\mathbf{r}'_S) - \frac{1}{2}\cdot\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right| - \eta\cdot\mathsf{X}_{\mathbf{r}}(\mathbf{r}'_S)\right] \leq \mathsf{p}_{\mathbf{r}} + \left(\frac{1}{2}-\eta\right)\cdot\eta$ holds for all $\mathbf{r}$, . We get

$$(\delta-\eta)\delta \leq \mathbb{E} \leq \frac{\delta^2}{2} + M\cdot\left[\eta/2 - \eta^2 + \max_{\mathbf{r}}\{\mathsf{p}_{\mathbf{r}}\}\right],$$

which rearranges to give $\max_{\mathbf{r}}\{\mathsf{p}_{\mathbf{r}}\} \geq \delta^2/3M = \delta^8/(2^{12}n\lambda^3)$ completing the proof of Claim 5. □

7 Sequential Non-malleability

In this section we prove Lemma 6 below. Together with Lemmas 4 and 5, this proves Theorem 4. We first set some notation.

Notation. Suppose a PPT MIM M plays two sequential executions of $\langle \mathrm{C}, \mathrm{R}\rangle$. First M plays on the left against C who commits honestly to m; let $\mathbb{T}_{\mathsf{L}}$ be the transcript of this interaction. Then M plays on the right against an honest R producing the transcript $\mathbb{T}_{\mathsf{R}} = \left(\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2, \tau_3, \{v_i\}_{i\in[n]}\right)$, where τ_2 and τ_3 contain the strings $\{\mathbf{r}_i, \mathbf{r}'_i\}_{i\in[n]}$. Let $\tilde{m}$ denote the committed value inside $\mathbb{T}_{\mathsf{R}}$. Let $\mathsf{MIM}^{\mathsf{M}}(m)$ denote the distribution which runs the above experiment and outputs $(\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}, \tilde{m})$. Proving non-malleability against M ammounts to showing that that for every non-negligible $\varepsilon > 0$, there exists a PPT simulator $\mathsf{SIM}^{\mathsf{M}}_{\varepsilon}$ which outputs $(\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}, \tilde{m})$ such that for all $m \in \{0,1\}^{\lambda}$, and polytime distinguishers D:

$$\left|\Pr_{\mathsf{MIM}^{\mathsf{M}}(m)}\left[\mathsf{D}(\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}, \tilde{m}) = 1\right] - \Pr_{\mathsf{SIM}^{\mathsf{M}}_{\varepsilon}}\left[\mathsf{D}(\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}, \tilde{m}) = 1\right]\right| \leq \varepsilon. \tag{5}$$

Lemma 6. *Assume* Com *is perfectly binding. Then for all non-negligible* $\varepsilon > 0$ *there exists a PPT simulator* $\mathsf{SIM}_{\varepsilon}$ *which, given oracle access to* M*, outputs* $(\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}, \tilde{m})$*. Moreover, for all PPT sequential MIMs, (5) holds.*

7.1 Proof Strategy

Fix a non-negligible $\varepsilon = \varepsilon(\lambda) > 0$. We describe a family of $N = \mathsf{poly}(\lambda)$ simulators $\mathsf{SIM}^{(1)}_{\varepsilon}, \ldots, \mathsf{SIM}^{(N)}_{\varepsilon}$ such that for one of the $\mathsf{SIM}^{(k)}_{\varepsilon}$, (5) holds for all $m \in \{0,1\}^{\lambda}$ and PPT sequential M. Each simulator $\mathsf{SIM}^{(k)}_{\varepsilon}$ is built using an extractor $\mathsf{NM.Ext}^{(k)}_{\varepsilon}$ which takes $\mathbb{T}_{\mathsf{R}}$ as input and outputs $\tilde{m}$. Given $\mathsf{NM.Ext}^{(k)}_{\varepsilon}$

and $m \in \{0,1\}^\lambda$, let $\mathsf{SIM}_\varepsilon^{(k)}(m)$ be the distribution which generates $(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m})$ as follows: 1) obtain $\mathbb{T}_\mathsf{L}$ by playing as an honest C committing to m in an execution of $\langle \mathrm{C}, \mathrm{R}\rangle$ against M; 2) generate $\mathbb{T}_\mathsf{R}$ by playing as an honest R in an execution of $\langle \mathrm{C}, \mathrm{R}\rangle$ against M; 3) run $\mathsf{Ext}_\varepsilon^{(k)}(\mathbb{T}_\mathsf{R})$ to get $\tilde{m}$. The simulator $\mathsf{SIM}_\varepsilon^{(k)}$ outputs a sample from $\mathsf{SIM}_\varepsilon^{(k)}(0^\lambda)$. The extractor $\mathsf{NM.Ext}_\varepsilon^{(k)}$ uses oracle access to M and also to $\{\mathsf{D}_\ell\}_{\ell<k}$ where each D_ℓ is a PPT distinguisher such that for some $m \in \{0,1\}^\lambda$,

$$\Pr_{\mathsf{MIM}^\mathsf{M}(m)}\big[\mathsf{D}_\ell(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1\big] > \Pr_{\mathsf{SIM}_\varepsilon^{(\ell)}}\big[\mathsf{D}_\ell(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1\big] + \varepsilon. \tag{6}$$

If no such D_ℓ exists for some $\ell < k$, then $\mathsf{SIM}_\varepsilon^{(\ell)}$ is the simulator we are looking for. The $\mathsf{NM.Ext}_\varepsilon^{(k)}$ are very similar to the extractor Ext_ε from Section 6. Like Ext_ε, the $\mathsf{NM.Ext}_\varepsilon^{(k)}$ make use of an inner extractor $\mathsf{NM.InnerExt}_\varepsilon$ and the message recovery subroutine $\mathsf{RecoverMsg}$ from Section 6.2.

Description of $\mathsf{NM.Ext}_\varepsilon^{(k)}$. Given $\mathbb{T}_\mathsf{R} = (\xi_1, \xi_2, \xi_3)$ as input and oracle access to M and $\{\mathsf{D}_\ell\}_{\ell<k}$:

1. If $\mathbb{T}_\mathsf{R}$ is not valid, output $\perp$; otherwise, initialize $\mathsf{FOUND} \subset [n] \times \{0,1\}^\lambda$ and $\mathsf{F} \subset [n]$ to $\emptyset$.
2. For all $\ell < k$, do the following:
   - **Inner Extraction Loop:** run $\mathsf{NM.InnerExt}_\varepsilon^{\mathsf{M},\mathsf{D}_\ell}(\xi_1, \mathsf{FOUND})$; if it outputs $(i, \mathbf{x}_i)$, update the sets $\mathsf{F} = \mathsf{F} \cup \{i\}$, $\mathsf{FOUND} = \mathsf{FOUND} \cup \{(i, \mathbf{x}_i)\}$; continue.
3. If $|\mathsf{FOUND}| \geq n - t$ for $t =???$, output $\mathsf{RecoverMsg}(\mathbb{T}_\mathsf{R}, \mathsf{FOUND})$; otherwise $\perp$.

The design and analysis of $\mathsf{NM.InnerExt}_\varepsilon$ are given in Section 7.2, but roughly speaking, the extraction guarantee is that whenever D_ℓ is such that (6) holds, $\mathsf{NM.InnerExt}_\varepsilon^{\mathsf{M},\mathsf{D}_\ell}(\xi_1, \mathsf{FOUND})$ has a good chance of recovering $(i, \mathbf{x}_i)$. If k is large enough, with high probability every $(i, \mathbf{x}_i)$ will be recovered. In this case, $\mathsf{SIM}_\varepsilon^{(k)}(m)$ and $\mathsf{MIM}^\mathsf{M}(m)$ are nearly identical, and so a distinguisher for $\mathsf{MIM}^\mathsf{M}(m)$ and $\mathsf{SIM}_\varepsilon^{(k)}$ breaks the hiding of $\langle \mathrm{C}, \mathrm{R}\rangle$. The formal proof of Lemma 6 is given in Section 7.3.

7.2 The Inner Extractor

The next claim enumerates the important properties of the inner extractor. The extractor construction and analysis are very similar to the inner extractor from Section 6.4.

Claim 6. *For all $\varepsilon > 0$ there exists an extractor $\mathsf{NM.InnerExt}_\varepsilon$ satisfying the following syntax, running time and extraction guarantees.*

- **Syntax:** $\mathsf{NM.InnerExt}_\varepsilon$ *is parametrized by $\varepsilon > 0$; it takes as input $\xi_1 = \big(\sigma_1, \tau_1, \{f_j, \mathbf{y}_j\}_{j\in[n]}\big)$ and a set*
$$\mathsf{FOUND} = \big\{(j, \mathbf{x}_j) : j \in \mathsf{F} \;\&\; f_j(\mathbf{x}_j) = \mathbf{y}_j\big\} \subset [n] \times \{0,1\}^\lambda$$
for some $\mathsf{F} \subset [n]$; gets oracle access to M and D and outputs either $(i, \mathbf{x}_i) \in [n] \times \{0,1\}^\lambda$ or $\perp$.
- **Running Time:** *The running time of $\mathsf{InnerExt}_\varepsilon$ is $\mathsf{poly}\big(\lambda, \mathsf{T}_\mathsf{M}, \mathsf{T}_\mathsf{D}, 1/\varepsilon\big)$.*
- **Extraction:** *Suppose (ξ_1, FOUND) with $|\mathsf{FOUND}| = n - \ell$ are such that $\ell > 0$ and either of the following hold:*
  - *$\ell > t$ and $\Pr_{\xi_2}\big[\mathbb{T}$ correct and $\mathbb{T}$ valid$\big|\xi_1\big] \geq \varepsilon/4$; or*

· $\ell \le t$ *and* $\Pr_{\xi_2}\big[\mathbf{E}\big|\xi_1\big] \ge \varepsilon/4$*,* $\mathbf{E}$ *the event "*$\mathbb{T}$ valid & F correct in $\mathbb{T}$*", and*

$$\Pr_{\xi_2}\big[\mathsf{D}(\mathbb{T}) = 1 \;\&\; \mathbb{T} \text{ correct}\big|\mathbf{E}\big] \ge 2^{-\ell} \cdot \Pr_{\xi_2}\big[\mathsf{D}(\mathbb{T}) = 1\big|\mathbf{E}\big] + \frac{\varepsilon}{2}.$$

Then with probability at least $1 - 2^{-\Omega(\lambda)}$*,* $\mathsf{NM.InnerExt}_\varepsilon^{\mathsf{M},\mathsf{D}}(\xi_1, \mathsf{FOUND})$ *outputs* $(i, \mathbf{x}_i)$ *such that* $i \notin \mathsf{F}$ *and* $f_i(\mathbf{x}_i) = \mathbf{y}_i$*.*

Construction of $\mathsf{NM.InnerExt}_\varepsilon$**.** Let $\varepsilon > 0$ be a parameter. Just as in the inner extractor of Section 6.4, this specifies $t = \mathcal{O}\big(\log 1/\varepsilon\big)$, $N' = \mathsf{poly}\big(\lambda, 1/\varepsilon\big)$, and $\eta = \mathsf{poly}\big(\lambda, \varepsilon\big)$.

• **Input:** $\mathsf{NM.InnerExt}_\varepsilon$ takes (ξ_1, FOUND) as input where $\xi_1 = \big(\sigma_1, \tau_1, \{f_j, \mathbf{y}_j\}_{j\in[n]}\big)$ and

$$\mathsf{FOUND} = \big\{(j, \mathbf{x}_j) : j \in \mathsf{F} \;\&\; f_j(\mathbf{x}_j) = \mathbf{y}_j\big\} \subset [n] \times \{0,1\}^\lambda,$$

for a set $\mathsf{F} \subset [n]$ of size $n - \ell$. $\mathsf{InnerExt}_\varepsilon$ gets oracle access to a sequential MIM M and a distinguisher D.

1. Let $\mathcal{I} = \big\{(i, S) : i \in T \;\&\; S \subset T \setminus \{i\}\big\}$ where $T \subset [n] \setminus \mathsf{F}$ is a set of size t if $\ell > t$; $T = [n] \setminus \mathsf{F}$ if $\ell \le t$.

2. For each $(i, S) \in \mathcal{I}$, do the following N' times or until $\mathbf{x}_i$ is recovered:

· Choose $\xi_2 = \big(\sigma_2, \tau_2\big)$ honestly, send ξ_2 to M and receive $\xi_3 = \big(\sigma_3, \tau_3, \{v_j\}_j\big)$. Let $\{\mathbf{r}_j, \mathbf{r}'_j\}_{j\in[n]}$ be the strings contained in (τ_1, τ_2, τ_3). Let $\mathbb{T} = (\xi_1, \xi_2, \xi_3)$; if $\mathbb{T}$ is not valid, continue.

· Define two GL-predictors, $\mathsf{Pred}_1, \mathsf{Pred}_2$, which take input $\mathbf{s} \in \{0,1\}^\lambda$, and work as follows:

(a) both predictors rewind M and sends $\hat{\xi}_2 = (\sigma_2, \hat{\tau}_2)$ where the only difference between $\hat{\tau}_2$ and τ_2 is the distribution of the strings $\{\hat{\mathbf{r}}'_j\}_{j\in[n]}$:

$$\hat{\mathbf{r}}'_j = \begin{cases} \mathbf{s}, & j = i \\ \mathbf{r}'_j, & j \in S \\ \text{random} \leftarrow \{0,1\}^\lambda, & j \notin S \cup \{i\} \end{cases}$$

(b) both predictors receive $\hat{\xi}_3 = (\hat{\sigma}_3, \hat{\tau}_3, \{\hat{v}_j\}_{j\in[n]})$ from C^* where $\hat{\tau}_3$ contains $\{\hat{\mathbf{r}}_j\}_j$, and output $\hat{v}_i$ if a set of checks pass, otherwise they output a random bit. The checks for Pred_1 are: 1) $\hat{\mathbb{T}} = (\xi_1, \hat{\xi}_2, \hat{\xi}_3)$ is valid; 2) $\hat{\mathbf{r}}_j = \mathbf{r}_j$ for all $j \in [n]$; 3) $\hat{v}_j = v_j$ for all $j \in S$; 4) $\hat{v}_j \oplus v_j = \langle \mathbf{x}_j, \mathbf{r}'_j \oplus \hat{\mathbf{r}}'_j \rangle$ for all $j \in \mathsf{F}$. Pred_2 uses checks (1)-(4) and also an additional 5) $\mathsf{D}(\hat{\mathbb{T}}) = 1$.

· Run the GL-inverter Inv_1 corresponding to Pred_1 and advantage η^2. If $\ell \le t$, run the GL-inverter Inv_2 corresponding to Pred_2 for advantage η^3. If either Inv_1 or Inv_2 recover $\mathbf{x}_i \in \{0,1\}^\lambda$ such that $f_i(\mathbf{x}_i) = \mathbf{y}_i$, output $(i, \mathbf{x}_i)$; otherwise continue.

3. This instruction is reached if the above loop has been executed N' times for each $(i, S) \in \mathcal{I}$ and nothing has ever been output; in this case output $\perp$.

Proof of Claim 6. It is clear that $\mathsf{NM.InnerExt}_\varepsilon$ has the required syntax for Claim 6. Moreover, the running time is $\mathsf{poly}\big(\lambda, 2^t, N', \mathsf{T}_\mathsf{M}, \mathsf{T}_\mathsf{D}, 1/\eta\big) = \mathsf{poly}\big(\lambda, \mathsf{T}_\mathsf{M}, \mathsf{T}_\mathsf{D}, 1/\varepsilon\big)$. Therefore, it suffices to establish the extraction guarantee. Suppose (ξ_1, FOUND) with $|\mathsf{FOUND}| = n - \ell$ is such that $\ell > t$ and

$\Pr_{\xi_2}\big[\mathbb{T}\text{ correct and valid}\big|\xi_1\big] \geq \varepsilon/4$. As Pred_1 is the same GL-predictor used by $\mathsf{InnerExt}_\varepsilon$, the proof of Claim 3 shows that $\mathsf{NM.InnerExt}_\varepsilon(\xi_1, \mathsf{FOUND})$ outputs $(i, \mathbf{x}_i)$ with high probability $1-2^{-\Omega(\lambda)}$. Suppose instead that (ξ_1, FOUND) is such that $\ell \leq t$, $\Pr_{\xi_2}\big[\mathbf{E}\big|\xi_1\big] \geq \varepsilon/4$, and

$$\Pr_{\xi_2}\big[\mathsf{D}(\mathbb{T}) = 1 \;\&\; \mathbb{T}\text{ correct}\big|\mathbf{E}\big] \geq 2^{-\ell} \cdot \Pr_{\xi_2}\big[\mathsf{D}(\mathbb{T}) = 1\big|\mathbf{E}\big] + \frac{\varepsilon}{2}.$$

It follows that there exists some $(i, S) \in \mathcal{I}$ and some main thread $\mathbb{T}$ generated during the loop for (i, S) such that the following hold for $\eta = \mathsf{poly}\big(\lambda, 1/\varepsilon\big)$:

(a) $\mathsf{F} \cup S$ is correct in $\mathbb{T}$ and $\mathbb{T}$ is valid;

(b) $\Pr_{\hat{\xi}_2}\big[\hat{\mathbf{E}}_S\big|\mathbf{r}'_S \in \hat{\xi}_2\big] \geq \eta$; and

(c) $\Pr_{\hat{\xi}_2}\big[\mathsf{D}(\hat{\mathbb{T}}) = 1 \;\&\; i\text{ correct in }\hat{\mathbb{T}}\big|\hat{\mathbf{E}}_S\big] \geq \frac{1}{2} \cdot \Pr_{\hat{\xi}_2}\big[\mathsf{D}(\hat{\mathbb{T}}) = 1\big|\hat{\mathbf{E}}_S\big] + \eta$;

where $\hat{\mathbf{E}}_S$ denotes the event: "$\hat{\mathbb{T}}$ is valid & $\mathsf{F} \cup S$ is correct in $\hat{\mathbb{T}}$ & the strings $\{\mathbf{r}'_j\}_{j\in S}$ and $\{\mathbf{r}_j\}_{j\in[n]}$ appear in $\hat{\xi}_2$ and $\hat{\xi}_3$", where $\{\mathbf{r}'_j\}_{j\in S}$ and $\{\mathbf{r}_j\}_{j\in[n]}$ are from $\mathbb{T}$. This follows from the arguments used to prove Claims 4 and 5. So assume points (a), (b) and (c) above all hold. We prove:

1. $\Pr_{\hat{\xi}_2}\big[\text{all checks pass}\big|\mathbf{r}'_S \in \hat{\xi}_2\big] \geq \eta^2$; and
2. $\Big|\Pr_{\hat{\xi}_2}\big[\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}_i\rangle\big|\text{all checks pass } \&\; \mathbf{r}'_S \in \hat{\xi}_2\big] - \frac{1}{2}\Big| \geq \eta$,

where the checks refer to the conditions checked by Pred_2. Together these imply that Pred_2 has advantage η^3 and so Inv_2 recovers $\mathbf{x}_i$ with high probability $1 - 2^{-\Omega(\lambda)}$ as desired. The probability on the left hand side of the first point is at least $\Pr_{\hat{\xi}_2}\big[\hat{\mathbf{E}}_S\big|\hat{\mathbf{r}}'_S \in \hat{\xi}_2\big] \cdot \Pr_{\hat{\xi}_2}\big[\mathsf{D}(\hat{\mathbb{T}}) = 1\big|\hat{\mathbf{E}}_S\big] \geq \eta^2$. For the second point, let $\mathsf{val} := \big|\Pr_{\hat{\xi}_2}\big[\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}_i\rangle\big|\text{all checks pass } \&\; \mathbf{r}'_S \in \hat{\xi}_2\big] - 1/2\big|$ be the quantity we want to bound. We have

$$\begin{aligned}
\mathsf{val} &= \left|\Pr_{\hat{\xi}_2}\Big[i\text{ correct in }\hat{\mathbb{T}}\Big|\hat{\mathbf{E}}_S \;\&\; \mathsf{D}(\hat{\mathbb{T}}) = 1\Big] - \frac{1}{2}\right| \\
&\geq \left|\Pr_{\hat{\xi}_2}\Big[i\text{ correct in }\hat{\mathbb{T}} \;\&\; \mathsf{D}(\hat{\mathbb{T}}) = 1\Big|\hat{\mathbf{E}}_S\Big] - \frac{1}{2} \cdot \Pr_{\hat{\xi}_2}\Big[\mathsf{D}(\hat{\mathbb{T}}) = 1\Big|\hat{\mathbf{E}}_S\Big]\right| \geq \eta.
\end{aligned}$$

□

7.3 Proof of Lemma 6

Proof. Fix non-negligible $\varepsilon = \varepsilon(\lambda) > 0$, a PPT sequential MIM M which breaks the non-malleability of $\langle \mathrm{C}, \mathrm{R}\rangle$, and let $N = \mathsf{poly}(\lambda) \in \mathbb{N}$ be such that $N > 12n^2/\varepsilon^2$. Let $\mathsf{SIM}_\varepsilon^{(1)}, \ldots, \mathsf{SIM}_\varepsilon^{(N)}$ be the simulators described in Section 7.1. Let $\mathsf{NM.Ext}_\varepsilon^{(k)}$ and D_k be the PPT extractor and PPT distinguisher corresponding to $\mathsf{SIM}_\varepsilon^{(k)}$. So

$$\Pr_{\mathsf{MIM}^{\mathsf{M}}(m)}\Big[\mathsf{D}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1 \;\&\; \tilde{m} \neq \bot\Big] \geq \Pr_{\mathsf{SIM}_\varepsilon^{(k)}}\Big[\mathsf{D}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1 \;\&\; \tilde{m} \neq \bot\Big] + \varepsilon,$$

holds for all $k = 1, \ldots, N$ (the $\tilde{m} \neq \bot$ clause is without loss of generality, since otherwise a trivial simulator which outputs $(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \bot)$ would simulate $\mathsf{MIM}^{\mathsf{M}}(m)$). Consider the following adversary $\mathcal{A}$ who plays the hiding game of $\langle \mathrm{C}, \mathrm{R}\rangle$:

1. $\mathcal{A}$ sends $(m_0, m_1) = (m, 0^\lambda)$ to $\mathcal{C}$ and receives $\mathbb{T}_\mathsf{L}$, a commitment to m_b ($\mathcal{A}$ plays as M for generating $\mathbb{T}_\mathsf{L}$). If $\mathbb{T}_\mathsf{L}$ is invalid, $\mathcal{A}$ outputs a random bit.
2. $\mathcal{A}$ generates $\mathbb{T}_\mathsf{R}$ by playing as M against an honest R in another execution of $\langle \mathrm{C}, \mathrm{R}\rangle$. If $\mathbb{T}_\mathsf{R}$ is invalid, $\mathcal{A}$ outputs a random bit.
3. $\mathcal{A}$ computes $\tilde{m} = \mathsf{NM.Ext}_\varepsilon^{(N)}(\mathbb{T}_\mathsf{R})$, except that $\mathcal{A}$ sets $\tilde{m} = \perp$ if $|\mathsf{FOUND}| < n$ by the end of the extractor execution.
4. Finally, if $\mathsf{D}_N(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1$, $\mathcal{A}$ outputs 0, otherwise $\mathcal{A}$ outputs a random bit.

Note $\mathcal{A}$ runs in time $\mathsf{poly}\big(\lambda, N, \mathsf{T}_\mathsf{M}, \max_k\{\mathsf{T}_{\mathsf{D}_k}\}, 1/\varepsilon\big) = \mathsf{poly}(\lambda)$. If $b = 1$, then $(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m})$ is a sample from $\mathsf{SIM}_\varepsilon^{(N)}$. We prove that $\tilde{m} = \mathsf{val}(\mathbb{T}_\mathsf{R})$ with probability at least $1 - \varepsilon/2$. It follows that that when $b = 0$, $(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m})$ is $\varepsilon/2-$close to a sample from $\mathsf{MIM}^\mathsf{M}(m)$, and so D_N has at least an $\varepsilon/2$ advantage in distinguishing $b = 0$ from $b = 1$. Thus $\mathcal{A}$ breaks the hiding of $\langle \mathrm{C}, \mathrm{R}\rangle$, so it remains to show that $\tilde{m} = \mathsf{val}(\mathbb{T}_\mathsf{L})$ with probability at least $1 - \varepsilon/2$.

So let $\mathbb{T}_\mathsf{R} = (\xi_1, \xi_2, \xi_3)$ and consider the execution of $\mathsf{NM.Ext}_\varepsilon^{(N)}(\mathbb{T}_\mathsf{R})$ which runs the inner extractor $\mathsf{NM.InnerExt}_\varepsilon^{\mathsf{M},\hat{\mathsf{D}}_k}(\xi_1, \mathsf{FOUND})$ for $k = 1, \ldots, N$, where $\hat{\mathsf{D}}_k(\mathbb{T}_\mathsf{R})$ outputs $\mathsf{D}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m})$, where $\tilde{m}$ is the decommitment of $(\sigma_1, \sigma_2, \sigma_3)$, the Com transcript in $\mathbb{T}_R$ (note $\mathsf{InnerExt}_\varepsilon^{\mathsf{M},\hat{\mathsf{D}}_k}$ only makes use of $\hat{\mathsf{D}}_k$ when $|\mathsf{FOUND}| \geq n - t$, in which case $\tilde{m}$ is available. Each time the inner extractor succeeds in recovering a one-way function preimage, a pair $(i, \mathbf{x}_i)$ is added to FOUND. Let $X_k := |\mathsf{FOUND}|$ after the $k-$th inner extractor execution. We think of X_k as a random variable with randomness ξ_1. Clearly, if all n OWF preimages are recovered (*i.e.*, when $X_N = n$), then $\mathsf{Ext}_\varepsilon^{(N)}(\mathbb{T}_\mathsf{R}) = \mathsf{val}(\mathbb{T}_\mathsf{R})$. Also, whenever $X_N < n$, $\mathsf{Ext}_\varepsilon^{(N)}(\mathbb{T}_\mathsf{R}) = \perp$. Therefore, it suffices to show that $\Pr\big[X_N < n \;\&\; \mathsf{val}(\mathbb{T}_\mathsf{R}) \neq \perp\big] \leq \varepsilon/2$. Say ξ_1 is *good* if $\Pr_{\xi_2}\big[\mathbb{T}_\mathsf{R} \text{ correct } \&\; \mathbb{T}_\mathsf{R} \text{ valid} \big| \xi_1\big] \geq \varepsilon/4$. Clearly, $\Pr\big[\mathsf{val}(\mathbb{T}_\mathsf{R}) \neq \perp \;\&\; \xi_1 \text{ not good}\big] < \varepsilon/4$. Thus, it suffices to show that $\mathsf{p} := \Pr_{\xi_1}\big[X_N < n \big| \xi_1 \text{ good}\big] < \varepsilon/4$.

Note the result follows from the expectation bound: $\mathbb{E}[X_{k+1}] \geq \min\big\{\mathbb{E}[X_k] + \varepsilon^2/12n, n - \varepsilon/4\big\}$ for all $k = 1, \ldots, N-1$, (expectations over good ξ_1). Indeed, this implies

$$n - \frac{\varepsilon}{4} \leq \mathbb{E}[X_N] \leq \mathsf{p} \cdot n + (1 - \mathsf{p}) \cdot (n - 1),$$

which rearranges to $\mathsf{p} \geq 1 - \varepsilon/4$, as desired (we have used $N > 12n^2/\varepsilon^2$). If $\Pr\big[X_k < n\big] < \varepsilon/4n$, then $\mathbb{E}[X_{k+1}]$ can be bounded directly: $\mathbb{E}[X_{k+1}] \geq \mathbb{E}[X_k] \geq n - \varepsilon/4$. Otherwise, let

$$\mathsf{q} := \Pr\big[\mathsf{NM.InnerExt}_\varepsilon^{\mathsf{M},\mathsf{D}_k} \text{ recovers a OWF preimage} \big| X_k < n\big].$$

Then, $\mathbb{E}[X_{k+1}] = \mathbb{E}[X_k] + \Pr\big[X_k < n\big] \cdot \mathsf{q} \geq \mathbb{E}[X_k] + \big(\varepsilon/4n\big) \cdot \mathsf{q}$, and so it suffices to show that $\mathsf{q} \geq \varepsilon/3$. Recall the probability in q is over good ξ_1. By Claim 6, $\mathsf{NM.InnerExt}^{\mathsf{M},\mathsf{D}_k}(\xi_1, \mathsf{FOUND})$ succeeds in extracting a OWF preimage with high probability $1 - 2^{-\Omega(\lambda)}$ whenever ξ_1 is good and either $|\mathsf{FOUND}| < n - t$, or $n - t \leq |\mathsf{FOUND}| < n$ and

$$\delta := \Pr_{\xi_2}\Big[\hat{\mathsf{D}}_k(\mathbb{T}_\mathsf{R}) = 1 \;\&\; \mathbb{T}_\mathsf{R} \text{ correct} \Big| \mathbf{E}\Big] - \Pr_{\xi_2}\Big[\hat{\mathsf{D}}_k(\mathbb{T}_\mathsf{R}) = 1 \Big| \mathbf{E}\Big] \geq \frac{\varepsilon}{2},$$

where $\mathbf{E}$ is shorthand for the event "$\mathbb{T}_\mathsf{R}$ is valid and F is correct in $\mathbb{T}_\mathsf{R}$". We have

$$\begin{aligned}
\varepsilon &\leq \Pr_{\mathsf{MIM}^\mathsf{M}(m)}\Big[\mathsf{D}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1 \;\&\; \tilde{m} \neq \perp\Big] - \Pr_{\mathsf{SIM}_\varepsilon^{(k)}}\Big[\mathsf{D}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1 \;\&\; \tilde{m} \neq \perp\Big] \\
&\leq \mathbb{E}_{\xi_1}\Big[\Pr_{\xi_2}\Big[\hat{\mathsf{D}}_k(\mathbb{T}_\mathsf{R}) = 1 \;\&\; \mathbb{T}_\mathsf{R} \text{ correct} \Big| \mathbf{E}\Big] - \Pr_{\xi_2}\Big[\hat{\mathsf{D}}_k(\mathbb{T}_\mathsf{L}, \mathbb{T}_\mathsf{R}, \tilde{m}) = 1 \Big| \mathbf{E}\Big]\Big] = \mathbb{E}_{\xi_1}[\delta].
\end{aligned}$$

Therefore, $\Pr_{\xi_1}\big[\delta \geq \varepsilon/2\big] \geq \varepsilon/2$, and so $\mathsf{q} \geq \big(\varepsilon/2\big) \cdot (1 - 2^{-\Omega(\lambda)}) \geq \varepsilon/3$, and we are done. $\square$

References

[ACJ17] Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A new approach to round-optimal secure multiparty computation. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 468–499, 2017.

[Bar02] Boaz Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '02, pages 345–355, 2002.

[BGJ+18] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 459–487, 2018.

[Blu81] Manuel Blum. Coin flipping by telephone. In *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981.*, pages 11–15, 1981.

[CCG+19] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. On round optimal secure multiparty computation from minimal assumptions. *IACR Cryptology ePrint Archive*, 2019:216, 2019.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 285–298, 2016.

[CGMO09] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position based cryptography. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 391–407, 2009.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, STOC '02, pages 494–503, 2002.

[COSV16a] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Concurrent non-malleable commitments (and more) in 3 rounds. In *CRYPTO*, 2016.

[COSV16b] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four Round Concurrent Non-malleable Commitments from One-way Functions. *CRYPTO*, 2017:621, 2016.

[COSV17a] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Delayed-input non-malleable zero knowledge and multi-party coin tossing in four rounds. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, pages 711–742, 2017.

[COSV17b] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Round-optimal secure two-party computation from trapdoor permutations. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, pages 678–710, 2017.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 670–683, 2016.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 542–552, 1991.

[DJMW12] Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. Counterexamples to hardness amplification beyond negligible. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 476–493, 2012.

[GKS16] Vipul Goyal, Dakshita Khurana, and Amit Sahai. Breaking the three round barrier for non-malleable commitments. *FOCS*, 2016.

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

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

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 695–704, 2011.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1128–1141, 2016.

[GRRV14] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *FOCS*, 2014.

[HHPV18] Shai Halevi, Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Round-optimal secure multi-party computation. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 488–520, 2018.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from Secure Multiparty Computation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, STOC '07, pages 21–30, 2007.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 158–189, 2017.

[Khu17] Dakshita Khurana. Round optimal concurrent non-malleability from polynomial hardness. *TCC*, 2017:734, 2017.

[KOS03] Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Round Efficiency of Multi-party Computation with a Dishonest Majority. In *Advances in Cryptology — EUROCRYPT '03*, volume 2656 of *Lecture Notes in Computer Science*, pages 578–595. Springer, 2003.

[KS17] Dakshita Khurana and Amit Sahai. Two-message non-malleable commitments from standard sub-exponential assumptions. *FOCS*, 2017:291, 2017.

[LP09] Huijia Lin and Rafael Pass. Non-malleability Amplification. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 189–198, 2009.

[LP11] Huijia Lin and Rafael Pass. Constant-round Non-malleable Commitments from Any One-way Function. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC '11, pages 705–714, 2011.

[LPS17] Huijia Lin, Rafael Pass, and Pratik Soni. Two-round concurrent non-malleable commitment from time-lock puzzles. *FOCS*, 2017:273, 2017.

[LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A Unified Framework for Concurrent Security: Universal Composability from Stand-alone Non-malleability. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 179–188, 2009.

[LS90] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 353–365, 1990.

[NSS06] Moni Naor, Gil Segev, and Adam Smith. Tight bounds for unconditional authentication protocols in the manual channel and shared key models. In *CRYPTO*, pages 214–231, 2006.

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

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive One-Way Functions and Applications. In *Advances in Cryptology — CRYPTO '08*, pages 57–74, 2008.

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

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 533–542, 2005.

[PW10] Rafael Pass and Hoeteck Wee. Constant-Round Non-malleable Commitments from Sub-exponential One-Way Functions. In *Advances in Cryptology — EUROCRYPT '10*, pages 638–655, 2010.

[Wee10] Hoeteck Wee. Black-Box, Round-Efficient Secure Computation via Non-malleability Amplification. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2010.

A Proof of Lemma 3

Proof. It is clear that $\mathsf{SIM}_{\mathsf{CF}}$ satisfies the syntax requirement. We now prove the running time and simulation requirements together using one hybrid argument, the coordinate hiding using another, and computational pairwise independence using a third.

Running Time and Simulation. We describe hybrid simulators.

- SIM_0**:** this simulator plays honestly with R^* and outputs the transcript $(\tau_1, \tau_2, \tau_3, \hat{\tau}_2, \hat{\tau}_3)$:

$$\Big(c, \{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\lambda]}, \{z_i\}_{i\in[n]}, \mathsf{rwi}_1; s', \{\mathbf{r}'_i\}_i, \mathsf{rwi}_2; \{\mathbf{r}_i\}_i, \mathsf{rwi}_3; \hat{s}', \{\hat{\mathbf{r}}'_i\}_i, \hat{\mathsf{rwi}}_2; \{\hat{\mathbf{r}}_i\}_i, \hat{\mathsf{rwi}}_3\Big),$$

  where $c = \mathrm{Com}(s \circ N_{\$})$ for random $s, N_{\$} \leftarrow \big[2^\lambda\big]$, $z^{\mathsf{GL}}_{b,\alpha} = \mathrm{Com}(\mathbf{r}^{\mathsf{GL}}_{b,\alpha})$ for random $\mathbf{r}^{\mathsf{GL}}_{b,\alpha} \leftarrow \{0,1\}^\lambda$, $z_i = \mathrm{Com}(\mathbf{r}^{\mathsf{hon}}_i)$ for random $\mathbf{r}^{\mathsf{hon}}_i \leftarrow \{0,1\}^\lambda$, $\mathbf{r}_i = \mathbf{r}^{\mathsf{hon}}_i$ for all i, and $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ is proved using the witness which involves the decommitment information for the z_i, but not c or the $z^{\mathsf{GL}}_{b,\alpha}$. The rewind messages are similar. Note, SIM_0 runs in time $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{R}^*}\big)$ and has output distribution identical to $\mathsf{REAL}^{\mathrm{R}^*}_{\mathsf{PairwiseCF}_n}$.

- SIM_1**:** this simulator has output identical to SIM_0; we make some procedural changes which mimic $\mathsf{SIM}_{\mathsf{CF}}$. Specifically, SIM_1 works as follows.

  – **Continue or Abort:** SIM_1 feeds R^* with an honest τ_1, if R^* aborts, SIM_1 outputs τ_1 and halts.

  – **Main Loop:** Otherwise, SIM_1 rewinds R^* repeatedly, sending independent honestly generated τ_1 to R^*, each time SIM_1 either receives τ_2 or R^* aborts. SIM_1 rewinds until R^* has responded with τ_2, $\lambda \cdot N$ times. Then SIM_1 picks one of the partial transcripts (τ_1, τ_2) at random and completes it to a full transcript (τ_1, τ_2, τ_3). Again, $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ is proved using the decommitment information of z_i only.

  – **Generate Rewind Transcript:** Feed R^* repeatedly with τ_1 until R^* responds with another valid $\hat{\tau}_2$ ($\hat{\tau}_2 = \tau_2$ is ok). Generate another response $\hat{\tau}_3$, as above.

  – **Output:** Output $(\tau_1, \tau_2, \tau_3, \hat{\tau}_2, \hat{\tau}_3)$.

  Note that the output of SIM_1 is identical to the output of SIM_0. To analyze the running time, let p denote the probability over τ_1 that R^* returns τ_2 instead of aborting. Note the probability that SIM_1 enters the main loop is p, and the expected time required to get through the main loop and to generate the rewind transcript is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{R}^*}, n, N\big)/\mathsf{p}$.

- SIM_2**:** this simulator is identical to SIM_1 except that in each τ_1 drawn during the main loop, the commitment $c = \mathrm{Com}(s \circ N)$ is prepared using the parameter N instead of $N_{\$} \leftarrow \left[2^{\lambda}\right]$ (s is still drawn randomly). Moreover, after SIM_2 has received $\lambda \cdot N$ responses τ_2, it aborts unless $s + s' \equiv 0 \pmod{N}$ holds for at least one of them. If continuing, SIM_2 chooses a τ_2 at random for which $s + s' \equiv 0 \pmod{N}$ holds and prepares τ_3 in the same way as SIM_1. Do the same for $\hat{\tau}_2$ and $\hat{\tau}_3$.

  Since the only difference between SIM_1 and SIM_2 are the committed values inside c used in the main loop, and the deocmmitment information is not used anywhere else in the protocol, the hiding of Com ensures that the running times of SIM_1 and SIM_2 are negligibly close to one another. Thus the expected runtime of SIM_2 is $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{R}^*}, n, N\big)$. Moreover, conditioned on R^* returning τ_2 instead of aborting, the probability that $s + s' \equiv 0 \pmod{N}$ holds is negligibly close to $1/N$. Again, this follows from the hiding of Com. Indeed, suppose this probability were noticeably greater than $1/N$. Then an adversary $\mathcal{A}$ could receive a commitment c to either $(s_0 \circ N)$ or $(s_1 \circ N)$ from outside, generate the remaining quantities itself and send τ_1 to R^*. Upon receiving s' as part of R^*'s response, $\mathcal{A}$ simply outputs 0 if $s_0 + s' \equiv 0 \pmod{N}$, and outputs a random bit if not, or if R^* aborts. Therefore, it is expected that of the $\lambda \cdot N$ times R^* responds with τ_2 instead of aborting, roughly λ of them will satisfy $s + s' \equiv 0 \pmod{N}$; the chance that none of them satisfy this constraint is $2^{-\Omega(\lambda)}$ by the Chernoff-Hoeffding inequality.

- SIM_3**:** This is the same as SIM_2 except that prior to beginning the main loop, SIM_3 fixes $j \in [n]$ arbitrarily and chooses $b \leftarrow \{0,1\}$. Then each τ_1 during the main loop is generated as follows: c is prepared as in SIM_2; $z_i = \mathrm{Com}(\mathbf{r}_i)$ for random $\mathbf{r}_i \leftarrow \{0,1\}^{\lambda}$; $z^{\mathsf{GL}}_{0,\alpha}$ and $z^{\mathsf{GL}}_{1,\alpha}$ are commitments to random $\mathbf{r}^{\mathsf{GL}}_{0,\alpha}, \mathbf{r}^{\mathsf{GL}}_{1,\alpha} \leftarrow \{0,1\}^{\lambda}$ such that $\mathbf{r}_j \in \mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{b,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{b,\mathsf{val}})$. The rewind values $\hat{\tau}_2, \hat{\tau}_3$ are generated as in SIM_2.

  The decommitment information for the $z^{\mathsf{GL}}_{0,\alpha}$ and $z^{\mathsf{GL}}_{1,\alpha}$ is not used in the rest of the protocol (the 'EITHER' witness is still used in the WI proof), so the hiding of Com immediately ensures that SIM_3 runs in expected time $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathrm{R}^*}, n, N\big)$ and has output indistinguishable from SIM_2. Note both the 'EITHER' and 'OR' parts of the statements proven are now true.

- SIM_4**:** This is the same as SIM_3 except that the 'OR' witness is used to complete the WI proofs in all transcripts generated during the main loop and during generation of $\hat{\tau}_2$ and $\hat{\tau}_3$. Witness indistinguishability against a rewinding adversary ensures SIM_4 is indistinguishable from SIM_3. Note, the output of SIM_4 is very similar to $\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(j)$; the only difference is that the committed value inside z_j is contained in $\mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{b,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{b,\mathsf{val}})$, rather than being uniformly random.

- SIM_5**:** This is the same as SIM_4 except that now z_j is a commitment to a random $\mathbf{r}_j \in \{0,1\}^{\lambda}$ instead of to a random string in $\mathsf{GL}(\mathbf{r}^{\mathsf{GL}}_{b,1}, \ldots, \mathbf{r}^{\mathsf{GL}}_{b,\mathsf{val}})$. The decommitment information of z_j is not used anywhere else in the protocol, so indistinguishability and running time follow immediately from the hiding of Com. Note that the distribution output by SIM_5 is identical to $\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(j)$. This establishes the running time and simulation guarantees of Lemma 3.

Computational Pairwise Independence. Fix PPTs R^* and D and $j \in [n]$. Let P_t denote the random variable which draws from $\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(1^{\lambda}, 1^n, 1^N)$ and outputs $\Pr_S\big[\mathsf{D}(\Sigma^j_{S,t}) = 1\big]$, probability over non-empty $S \subset \{1, \ldots, \mathsf{val}\}$. We prove that $\mathbb{E}_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\left[\mathsf{P}^2_t - \mathsf{P}_t \cdot \mathsf{P}_s\right] < 1/2\lambda^2 N^3$ for all $s, t \in [\lambda]$. The bound

(3) follows via Markov's inequality:

$$\Pr_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\Big[\big|\mathsf{P}_t - \mathsf{P}_s\big| \geq 1/N\Big] \leq N^2 \cdot \mathbb{E}_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\Big[\mathsf{P}_t^2 + \mathsf{P}_s^2 - 2\mathsf{P}_t\mathsf{P}_s\Big] < \frac{1}{\lambda^2 N}.$$

Note $\mathsf{P}_t = (2^{\mathsf{val}} - 1)^{-1} \cdot \sum_S \mathbb{1}_{S,t}$ where the sum is over non-empty $S \subset \{1, \ldots, \mathsf{val}\}$ and where $\mathbb{1}_{S,t} = \mathsf{D}(\Sigma^j_{S,t})$ is shorthand. It follows that

$$\mathbb{E}_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\Big[\mathsf{P}_t^2 - \mathsf{P}_t\mathsf{P}_s\Big] < 2^{-\mathsf{val}} + \max_{S \neq T}\Big\{\mathbb{E}_{\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}}\Big[\mathbb{1}_{S,t} \cdot \mathbb{1}_{T,t} - \mathbb{1}_{S,s} \cdot \mathbb{1}_{T,t}\Big]\Big\},$$

and so it suffices (since $\mathsf{val} = 2\log(\lambda) + 3\log(N) + 2$) to show that $\mathcal{S}(S,t,T,t) \approx_{\mathsf{c}} \mathcal{S}(S,s,T,t)$ for all $s,t \in [\lambda]$ and non-empty $S \neq T \subset \{1,\ldots,\mathsf{val}\}$ where $\mathcal{S}(S,s,T,t)$ denotes the distribution which draws from $\mathsf{SIM}^{\mathrm{R}^*}_{\mathsf{CF}}(1^\lambda, 1^n, 1^N)$ and outputs $\big(\tau_1, \tau_2, \tau_3(j, \overline{\mathbf{r}}_{S,s}), \hat{\tau}_2, \hat{\tau}_3(j, \overline{\mathbf{r}}_{T,t})\big)$. For this we use one final hybrid argument.

- H_0**:** this hybrid outputs a sample from $\mathcal{S}(S,t,T,t)$. Specifically, it outputs

  $$\big(\tau_1, \tau_2, \tau_3(j, \overline{\mathbf{r}}_{S,t}), \hat{\tau}_2, \hat{\tau}_3(j, \overline{\mathbf{r}}_{T,t})\big).$$

  Just as in G_0, τ_1 contains commitments $\{z^{\mathsf{GL}}_{0,\alpha}, z^{\mathsf{GL}}_{1,\alpha}\}_{\alpha\in[\mathsf{val}]}$ and $\{z_i\}_{i\in[n]}$, to strings $\mathbf{r}^{\mathsf{GL}}_{b,\alpha}$ where for a random $b \in \{0,1\}$, the committed string inside $z^{\mathsf{GL}}_{b,\alpha}$ is $\mathbf{r}^{\mathsf{GL}}_\alpha$ and where $\overline{\mathbf{r}}_{S,t} = \mathbf{e}_t \oplus \big(\bigoplus_{\alpha\in S} \mathbf{r}^{\mathsf{GL}}_\alpha\big)$, and likewise for $\overline{\mathbf{r}}_{T,t}$. The decommitments of the $z^{\mathsf{GL}}_{1-b,\alpha}$ do not appear in the output of H_0.

- H_1**:** this is the same as H_0 except that the committed strings $\mathbf{s}^{\mathsf{GL}}_\alpha$ inside the $z^{\mathsf{GL}}_{1-b,\alpha}$ are drawn randomly such that

  $$\mathbf{e}_t \oplus \big(\bigoplus_{\alpha\in S} \mathbf{r}^{\mathsf{GL}}_\alpha\big) = \mathbf{e}_s \oplus \big(\bigoplus_{\alpha\in S} \mathbf{s}^{\mathsf{GL}}_\alpha\big); \text{ and } \mathbf{e}_t \oplus \big(\bigoplus_{\beta\in T} \mathbf{r}^{\mathsf{GL}}_\beta\big) = \mathbf{e}_t \oplus \big(\bigoplus_{\beta\in T} \mathbf{s}^{\mathsf{GL}}_\beta\big).$$

  Note that since the decommitment information of the $z^{\mathsf{GL}}_{1-b,\alpha}$ are not used to complete the RWI proof, $\mathsf{H}_1 \approx_{\mathsf{c}} \mathsf{H}_0$ follows immediately from the hiding of Com. Note that the $1-b$ witness is now active.

- H_2**:** this is the same as H_1 except that now the $1-b$ witness is used to complete the proofs, instead of the b witness. $\mathsf{H}_2 \approx_{\mathsf{c}} \mathsf{H}_1$ follows immediately from the $2-$rewind secure witness indistinguishability of RWI. Notice now that the decommitments of the $z_{b,\alpha}$ do not appear in the output of H_2.

- H_3**:** This hybrid is the same as H_2 except that the commitment strings inside the $z_{b,\alpha}$ are chosen randomly. $\mathsf{H}_3 \approx_{\mathsf{c}} \mathsf{H}_2$ follows from the hiding of Com. Since the samples output by H_3 are identical to $\mathcal{S}(S,s,T,t)$, so computational pairwise independence is established.

□

B Non-Malleability Against a Synchronizing MIM

In this section we discuss the ideas behind the proof of the following lemma. Roughly speaking, the lemma is proved using the argument from [GPR16] and the proof of hiding from Section 5.

Lemma 7. *Assume that one-to-one one-way functions exist, let* Com *be the main component of the commitment scheme from [GPR16], and let* $\mathsf{PairwiseCF}_n$ *be a pairwise independent coinflipping protocol. Then* $\langle \mathrm{C}, \mathrm{R} \rangle$ *is non-malleable against a synchronizing MIM.*

Notation. In this section, a synchronizing MIM plays two executions of $\langle \mathrm{C}, \mathrm{R} \rangle$ obtaining a transcript

$$\mathbb{T} = \big(\sigma_1, \xi_1; \sigma_1', \xi_1'; \sigma_2', \xi_2'; \xi_2, \sigma_2; \xi_3, \sigma_3; \xi_3', \sigma_3'\big),$$

where $(\sigma_1, \sigma_2, \sigma_3)$ and $(\sigma_1', \sigma_2', \sigma_3')$ are the GPR components on the left and right executions, respectively; and where $(\xi_1, \xi_2, \xi_3) = \big(\tau_1, \{f_i, \mathbf{y}_i\}; \tau_2; \tau_3, \{v_i\}\big)$ are the remaining left messages and likewise (ξ_1', ξ_2', ξ_3') are the remaining right messages. The ordering of the messages in the tuple $\mathbb{T}$ above represents the chronological ordering in which the messages would appear when M is a synchronizing MIM. Typically, for notational simplicity, we will write $\mathbb{T} = \big(\sigma_1, \xi_1; \sigma_2', \xi_2'; \sigma_3, \xi_3\big)$, since the remaining messages are implicitly defined from these as they are outputs of M.

Synchronizing NM of the GPR Main Component. We begin by recalling the high level proof of synchronizing NM when the main component from GPR is played by itself. In this case, only the σ messages appear in $\mathbb{T}$. The GPR main component used a non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ as a subroutine and the proof of synchronizing NM was by reduction to the non-malleability of the code. The observation was that one could use the data $\mathsf{params} := (\sigma_1, \sigma_2', \sigma_3, \hat{\sigma}_2')$ to define a function f_{params} mapping $\mathsf{Enc}(m)$ to $\mathsf{Enc}(m')$ (m' the committed value in the right execution). The params consist of a synchronizing transcript $\mathbb{T} = (\sigma_1, \sigma_2', \sigma_3)$ and a partial rewind transcript $(\sigma_1, \hat{\sigma}_2')$. The function f_{params} takes $\mathsf{Enc}(m)$ as input, generates $\hat{\sigma}_3$ so completing the partial rewind transcript to a full rewind transcript, and recovers $\mathsf{Enc}(m)$ from $(\sigma_1', \sigma_2', \sigma_3', \hat{\sigma}_2', \hat{\sigma}_3')$. The assumption that M mauls $\langle \mathrm{C}, \mathrm{R} \rangle$ means that there exists $m \in \{0,1\}^\lambda$, a PPT distinguisher D and non-negligible $\varepsilon > 0$ such that

$$\Pr_{\mathsf{params}}\Big[\Pr\Big[\mathsf{D}\big(\mathsf{Tamper}^m_{f_{\mathsf{params}}}\big) = 1\Big] \geq \Pr\Big[\mathsf{D}\big(\mathsf{Tamper}^{0^\lambda}_{f_{\mathsf{params}}}\big) = 1\Big] + 2\varepsilon\Big] \geq 2\varepsilon, \tag{7}$$

where Tamper^m_f is the distribution $(\mathsf{Dec} \circ f \circ \mathsf{Enc})(m)$. As written, it looks as though (7) means that the non-malleability of the code is broken. However, the function f_{params} does not belong to the class of functions that $(\mathsf{Enc}, \mathsf{Dec})$ is secure against. Essentially, the problem is that f_{params} depends on some of the random choices made during computing $\mathsf{Enc}(m)$ as some of these choices appear in the transcript $(\sigma_1, \sigma_2, \sigma_3)$. This problem is overcome by exhibiting an indistinguishable distribution $\mathcal{D}$ on params where for all $\mathsf{params} \in \mathrm{Supp}(\mathcal{D})$, $f_{\mathsf{params}} \in \mathcal{F}$, the family of tampering functions that $(\mathsf{Enc}, \mathsf{Dec})$ is non-malleable against. It is then shown that

$$\Pr_{\mathsf{params} \leftarrow \mathcal{D}}\Big[\Pr\Big[\mathsf{D}\big(\mathsf{Tamper}^m_{f_{\mathsf{params}}}\big) = 1\Big] \geq \Pr\Big[\mathsf{D}\big(\mathsf{Tamper}^{0^\lambda}_{f_{\mathsf{params}}}\big) = 1\Big] + \varepsilon\Big] \geq \varepsilon \tag{8}$$

holds, which does break the non-malleability of $(\mathsf{Enc}, \mathsf{Dec})$. Proving (8) uses a hybrid argument. The idea is that since D, f_{params} and the original sampling procedure of params are all polytime, if (7) holds but (8) does not, then we can break the security of some cryptographic subroutine.

Synchronizing NM of $\langle \mathrm{C}, \mathrm{R} \rangle$. The proof that $\langle \mathrm{C}, \mathrm{R} \rangle$ is non-malleable against a synchronizing MIM is very similar. Essentially the only difference from the above is that params includes more information:

$$\mathsf{params} = \big(\sigma_1, \tau_1, \{f_i, \mathbf{y}_i\}; \sigma_2', \tau_2'; \sigma_3, \tau_3, \{v_i\}; \hat{\sigma}_2', \hat{\tau}_2'; \hat{\tau}_3, \{\hat{v}_i\}\big).$$

In addition to requiring a transcript and the second message of a rewind transcript, params requires the non-GPR part of the third message of the rewind transcript. These allow f_{params} to use $\mathsf{Enc}(m)$ to generate $\hat{\sigma}_3$ and get a full rewind transcript to recover $\mathsf{Enc}(m')$. As above, the idea is to describe an

indistinguishable distribution on params so that $f_{\mathsf{params}} \in \mathcal{F}$ breaks the non-malleability of $(\mathsf{Enc}, \mathsf{Dec})$. Note that $\{v_i\}$ and $\{\hat{v}_i\}$ in params are generated as $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{s}_i \rangle \oplus c_i$ and $\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}_i \oplus \hat{\mathbf{s}}_i \rangle \oplus c_i$, where $\mathbf{s}_i, \mathbf{r}_i, \hat{\mathbf{s}}_i, \hat{\mathbf{r}}_i$ are the strings contained in $\tau_2, \tau_3, \hat{\tau}_2, \hat{\tau}_3$ (we have changed $\mathbf{r}'_i$ to $\mathbf{s}_i$ here so that primes only appear on messages of the right execution); $\mathbf{x}_i$ is such that $f_i(\mathbf{x}_i) = \mathbf{y}_i$, and $(c_1, \ldots, c_n)$ is the GPR decommitment information in σ_1. We change the distribution of params in two phases. First, we change params so that $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{s}_i \rangle$ and $\hat{v}_i = \langle \mathbf{x}_i, \hat{\mathbf{r}}_i \oplus \hat{\mathbf{s}}_i \rangle$ (*i.e.*, we remove the dependence on $\mathrm{Decom}(\sigma_1, \sigma_2, \sigma_3)$ from $\{v_i\}$ and on $\mathrm{Decom}(\sigma_1, \hat{\sigma}_2, \hat{\sigma}_3)$ from $\{\hat{v}_i\}$). Then we change the GPR part of params to being drawn from $\mathcal{D}$ as above. Just as in the proof of hiding in Section 5, if the first change is noticeable, then an efficient adversary can be constructed to invert some f_i.