Non-Malleable Commitments using Goldreich-Levin List Decoding

Vipul Goyal
Carnegie Mellon University
Email: goyal@cs.cmu.edu

Silas Richelson
UC Riverside
Email: silas@cs.ucr.edu

***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 an ϵ-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.

***Keywords*-cryptographic protocols, non-malleable commitments, Goldreich-Levin Decoding.**

I. 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 that 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

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00047

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?*

A. 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 ϵ-extractable commitment scheme from injective one-way functions. Our scheme does not suffer from "over-extraction", or, "under-extraction". Very roughly, this means that if the committer is committing to $\perp$, the extractor must not output a valid value, and vice-versa (please see the next Section for details). Our scheme features an extractor which, for any ϵ, extracts the committed value with probability at least $1-\epsilon$ by running in time poly$(1/\epsilon)$. We stress that the protocol, however, does not fix a specific value of ϵ. Our scheme is secure as per the specific definition given in Section II. The only previous such construction was due to Jain et al [JKKR17] and relied on specific number theoretic assumptions (namely either DDH or QR residuosity). Our techniques are essentially unrelated to those used in [JKKR17]. In particular, we introduce what we call *bidirectional Goldreich-Levin style extraction*. Very roughly, this allows both sides of the security proof (i.e., when the committer is corrupt as well as when the receiver is corrupt) to rely on inverting the injective one-way function by using inner products produced by the corrupted party. We believe this to be of independent interest.

A crucial building block we construct and use in our work is that of a delayed-input rewind secure witness-indistinguishable (WI) proof in 3-rounds from injective one-way functions. Very informally, 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). Goyal et al [GRRV14] designed an additional secret sharing technique to ensure that this was enough for their purposes. 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, ϵ-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.

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 by Choudhuri, Goyal, and, Jain [CGJ18] to obtain a construction 4-round MPC from only injective one-way functions and oblivious transfer. 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.

B. Technical Overview

Our main technical tool is the construction of a 3-round extractable commitment scheme for injective one-way functions which does not suffer from "over-extraction", or, "under-extraction". Very roughly, this means that if the committer is committing to $\perp$, the extractor must not

[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.

output a valid value, and vice-versa. Consider the commonly used three round extractable commitment scheme where the committer C uses XOR secret sharing to break the secret into two shares (called a pair of shares). C prepares k such pair of shares and commits for each of them. The receiver R then chooses one share from each pair at random in the second round. The selected shares are then decommitted in the final round. Indeed this scheme is extractable since the extractor can recover both the shares for at least one pair. However this scheme suffers from over-extraction. Say a cheating committer C^* prepares the pairs such that not every pair leads to the same secret (and hence the committed string is $\perp$). Even in this case, the extractor may still extract and output a valid value. This problem can be solved by using a zero-knowledge proof of consistency which would then require an additional round. We refer to the extractable commitment schemes which suffer from overextraction or underextraction as weakly extractable commitment schemes. Weakly extractable commitment are not sufficient for our purposes since they lead to subtle "selective $\perp$" attacks as far as mauling attacks are concerned.

We highlight our main ideas by going through various attempts to build such a scheme. Our commitment scheme is delayed input and the message is only required by C in the last round. Assume for simplicity that the message to be committed to is a single bit. Furthermore, assume that C does not simply sample and commit a random input bit in the last round. That is, there exists a bit m s.t. C commits to m with probability at least $\frac{1}{2}+\epsilon$ (where ϵ is noticeable in the security parameter) across the different rewound executions by the extractor.

Background on the Goldreich-Levin Theorem: Here we give an informal overview of the influential Goldreich-Levin theorem [GL89] which is relevant to the following discussion. An expanded background and formal details can be found in Section II-C. The main result in [GL89] shows that every one-way function has a hardcore predicate. The core of their proof is the "prediction implies inversion" lemma, which shows roughly that if an algorithm can, given $f(x)$, predict random inner products of x with probability at least $1/2+\epsilon$ (where ϵ is noticeable in the security parameter), then this prediction algorithm can be used to invert f and recover x. We refer to this inversion algorithm as the Goldreich-Levin algorithm.

The Goldreich-Levin algorithm queries the inner product on correlated strings coming from a special distribution. The strings are generated as follows. First sample uniform strings $\mathbf{r}_1,\ldots,\mathbf{r}_k \in \{0,1\}^\lambda$ (where λ denotes the security parameter). For 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. Now we define $\mathsf{GL}(\mathbf{r}_1,\ldots,\mathbf{r}_k)$ to be the following set:

$$\left\{\mathbf{r}_{S,i} \in \{0,1\}^\lambda : \emptyset \neq S \subset \{1,\ldots,k\}, i \in \{1,\ldots,\lambda\}\right\}.$$

We refer to the strings in this set as GL queries. Notice that the number of GL queries is exponential in k. k is required to be only logarithmic assuming ϵ is noticeable in the security parameter. The value of k and hence the number of GL queries depends upon the value of ϵ.

The Goldreich-Levin algorithm requires the output of the prediction algorithm on all GL queries, and, is guaranteed to output x with noticeably probability. The set of queries in $\mathsf{GL}(\mathbf{r}_1,\ldots,\mathbf{r}_k)$ is not uniform. However, Goldreich and Levin [GL89] were able to exploit the fact that the queries are *pairwise independent*.

The Starting Protocol: Our starting point is the following basic non-interactive commitment scheme based on any injective OWF f. The committer C samples random strings x, r, and, sends $f(x), \langle x, r\rangle \oplus m$ as a commitment to the bit m. To make this commitment scheme extractable, we modify it and let the receiver R choose r. Hence, C sends $f(x)$ in the first round, R responds with r, and, C sends $\langle x, r\rangle \oplus m$ in the last round. It can be shown that this scheme is extractable. The extractor Ext can rewind C and get the inner products for multiple strings $(r_1, r_2, \ldots)$. Ext can simply guess the bit m (or go over both possibility for m) and recover the inner products $(\langle x_1, r\rangle, \langle x_2, r\rangle, \ldots)$. If indeed C was committing to m at least with probability $\frac{1}{2}+\epsilon$, each of the inner products is correct with probability at least $\frac{1}{2}+\epsilon$. Then, by using the Goldreich-Levin algorithm, Ext can now recover x and hence recover the committed bit.

The hiding of this protocol is however unclear. Indeed, this protocol is unlikely to satisfy the hiding property since a cheating receiver R^*, given $f(x)$, maybe able to guess $\langle x, r\rangle$ for a specific string r even possibly with probability 1. By using this string r in the second round, R^* would be able to recover the committed bit.

Bidirectional Goldreich-Levin: Suppose we could construct a single protocol which satisfies the following two properties:

1) If C is corrupt, it is possible for Ext to choose r. This would allow for successful extraction as explained before.
2) If R is corrupt, its however possible for the inverter Inv to choose r instead. Recall that our proof of hiding would work by constructing an inverter for the OWF f (given an adversary adv to break the hiding property). In that case, it would be possible for Inv to recover x as well by using the Goldreich-Levin algorithm. Inv would take $f(x)$ externally and would simply give a random bit b to adv in place of $\langle x, r\rangle \oplus m$. advwould now respond with a guess for the message m which would then allow the inverter to recover a guess for $\langle x, r\rangle$. Indeed, if Inv could control r and adv guesses the message correct with probability at least $\frac{1}{2}+\epsilon$, Inv could rewind adv to recover several guesses $(\langle x_1, r\rangle, \langle x_2, r\rangle, \ldots)$ each correct with probability at least $\frac{1}{2}+\epsilon$. Inv could then run the Goldreich-Levin algorithm to recover x thus contradicting the security of f.

Indeed, the question now is about designing a protocol where r could be controlled by Ext as well as Inv.

Using Coin Flipping: A natural next idea is to try coin flipping to generate r. Consider the following protocol. C sends $f(x)$ and $\mathsf{Com}(r)$ where x, r are sampled uniformly and Com is a non-interactive perfectly binding commitment scheme. The receiver R responds with a uniform r'. Finally, C opens r and sends $\langle x, r \oplus r' \rangle \oplus m$. The extraction continue to work in this case since Ext can simply rewind C to learn r, and, continue to learn inner products on strings $(r_1, r_2, \dots)$ of its choice simply by sending $(r_1 \oplus r', r_2 \oplus r', \dots)$ in the second round.

The proof of hiding still remains unclear however. Indeed, Inv can rewind R and commit to a fresh random string r. However, Inv has no control over r' chosen by R (which may be different in different rewinds). Hence, the inverter can recover the inner products $(\langle x_1, r \rangle, \langle x_2, r \rangle, \dots)$ for random strings $(r_1, r_2, \dots)$ which are *beyond its control*. However recall that in order to be successful, the Goldreich-Levin algorithm needs to get inner products for carefully chosen strings which are correlated with each other.

A plausible solution here would be to use a two-sided *simulatable* coin flipping which would allow Inv to fully control the strings $(r_1, r_2, \dots)$ by controlling the outcome of the coin flipping protocol. However, this would require an additional round of interaction.

Committing Several Strings in the First Round: We now consider the following protocol. C sends $f(x)$ and $\mathsf{Com}(r_1), \dots, \mathsf{Com}(r_\ell)$ where $x, r_1, \dots, r_\ell$ are sampled uniformly. R responds with r' as before. Finally, C chooses a random $i \in [\ell]$, opens $\mathsf{Com}(r_i)$, and, sends $\langle x, r_i \oplus r' \rangle \oplus m$.

The extraction can still be made to work by relying on the fact that ℓ is polynomial in the security parameter. The extractor runs the protocol once. Suppose C chooses to open $r_i, i \in [\ell]$ in the third round. Now Ext starts rewinding C and focuses only on transcripts where C chooses to open the same string r_i in the third round. If C does not choose r_i, Ext simply rewinds and tries again with a fresh r'. This would allow Ext to follow the same strategy as in the previous case where C was committing to a single r in the first round.

Now consider the following strategy for Inv. Inv takes $f(x)$ as an external input and prepares $\mathsf{Com}(r_1), \dots, \mathsf{Com}(r_\ell)$ honestly. It completes the execution till the second round using these values and receives r'. Inv then prepares a guess for $\langle x, r' \rangle$. Now execute the third round multiple times by rewinding adv and opening $r_1, \dots, r_\ell$ respectively. In each rewind, Inv simply sends a random bit in place of $\langle x, r_i \oplus r' \rangle \oplus m$. The response of adv would allow Inv to obtain guesses for $\langle x, r_1 \oplus r' \rangle, \dots, \langle x, r_\ell \oplus r' \rangle$. If its guess for $\langle x, r' \rangle$ is correct, Inv can now obtained guesses for $\langle x, r_1 \rangle, \dots, \langle x, r_\ell \rangle$. Now since Inv has complete over the choice of $r_1, \dots, r_\ell$, this allows Inv to execute the Goldreich-Levin algorithm on the GL queries and recover x to arrive at a contradiction.

While it seems like we are making progress, we now run into the following problem. Suppose advpredicts the bit committed bit with probability $\frac{1}{2} + \epsilon$. The number of GL queries ℓ required for the Goldreich-Levin algorithm to succeed would depend upon ϵ. Note that ϵ is unknown during the protocol execution, and hence, its unclear how many strings C must commit to in the first round (i.e., no fixed polynomial ℓ would suffice). Our inverter indeed is given ϵ as input and hence can decide how many strings to commit to. However the inverter cannot freely decide the number of committed strings since ℓ is fixed by the protocol description.

Using Implicit Representation of GL Queries: We now observe the following. The ℓ GL queries can actually be implicitly represented by a much smaller set of string from which all GL queries can be generated (see the background on Goldreich-Levin above). Our next idea is to only commit to a fixed polynomial number of strings $(r_1, \dots, r_\lambda)$ in the first round (in fact, any super-logarithmic number of strings will suffice). In the last round, C sends a string r sampled uniformly from the set $\mathsf{GL}(r_1, \dots, r_\lambda)$, and, proves using a 3 round WI protocol that r is indeed in this set (more on the WI protocol later).

This now allows the inverter Inv to rewind adv and issue any (unbounded) polynomial number of queries where the polynomial is not a priori fixed by the protocol. This would allow Inv to overcome the previous issue and handle any noticeable ϵ.

However the problem now occurs in extraction. The size of the set $\mathsf{GL}(r_1, \dots, r_\lambda)$ is now super-polynomial. Hence, the string r which C chooses in the last round can no longer be predicted with noticeable probability. If the extractor "focusses" on a single r, it could potentially start running in super-polynomial time.

Using Unbounded Polynomial Commitments: Our next idea to use a special commitment scheme which allows the committer to attempt to commit to any value $N \in 2^\lambda$. However, the committer will only be successful in committing to N with probability approximately $\frac{1}{N}$, and, otherwise the committed value will be 1. R would not know what the committed value is (including whether it is to 1 or not), but, the committer will. We construct such a commitment scheme in Section III-A.

Now consider the following protocol. C, in additional to committing to $(r_1, \dots, r_\lambda)$, also samples a random $N \in [2^\lambda]$ and attempts to commit to N using the unbounded polynomial commitment scheme. If C is successful in committing to N, define $k = \lceil log(N) \rceil$, and, $k = 1$ if C instead commits to 1. In the lat round, C would prove (in WI) that the given string s, in fact, belongs to the set $\mathsf{GL}(r_1, \dots, r_k)$. Thus, while C commits to $(r_1, \dots, r_\lambda)$, the number of strings which are "active" is determined by the value inside unbounded polynomial commitment.

The proof of extraction now starts to work. W.h.p, C^*

is only able to commit to N s.t. N is polynomial in the security paramter λ. Thus, the number of active strings k is logarithmic, and hence, the space of possibilities for r in the last round is bounded by a polynomial. This again allows the extractor to focus on a specific r in order to extract x.

The proof of hiding seems to work as well (at least at a high level). Given ϵ, Inv computes the number of required GL queries and hence the number of active strings k. It sets $N = 2^k$. Note that N is still guaranteed to be a polynomial. Inv now attempts to commit to N and simply tries again by rewinding R if it is unsuccessful. By trying approximately N times in expectation, Inv will be successful and hence will now be able to get the inner product guesses on sufficient number of strings to be able run the Goldreich-Levin algorithm.

Using Rewind Secure WI: Given the above ideas, the proof of hiding still unfortunately does not work because of a technical issue arising from the pairwise independence requirement in the Goldreich-Levin theorem. The pairwise independence requirement seems to translate to, very roughly, "pairwise independence of the views" in our setting (with a fixed first message). Indeed, our protocol is only computational and such a property would not hold for an unbounded adversary. We are able to work with a computational notion of pairwise independence which still allows for the Goldreich-Levin style probability analysis to go through. However, this notion requires the following. Even given last round messages from two different executions with the same first message (i.e., two different executions resulting from rewinding adv), adv cannot distinguish if the two queries are of the form $(\mathbf{r}_{S,i}, \mathbf{r}_{T,i})$, or, $(\mathbf{r}_{S,i}, \mathbf{r}_{T,j})$ with $i \neq j$.

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 the WI protocol are off. To solve this problem, we use a *delayed input rewind secure WI* one 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 as well. Our key technique relies on using "two-layers" of MPC in the head technique of Ishai et. al [IKOS07] along with a careful combinatorial analysis. More details are given in Section III-B.

Finally, we note that for WI to provide any meaningful guarantees, there must exist at least two witness for the statement being proven. Towards that end, very roughly, we actually run two parallel copies of the building blocks (including the implicit commitments to the GL queries, and, the unbounded polynomial commitment) discussed so far. We are able to use the two copies in conjunction with a careful hybrid argument to achieve our computational equivalent of the pairwise independence property.

Going to Non-Malleable Commitments: The basic protocol from GPR is non-malleable against a synchronizing adversary. However non-malleability fails against a sequential adversary (which completes the left session before starting the right session), essentially because their scheme is not extractable. By composing their protocol with our three round $\varepsilon-$extractable commitment scheme in parallel, we obtain non-malleability against a sequential adversary due to the extraction properties of our additional scheme. The key point is that $\varepsilon-$extractability suffices for proving non-malleability since extraction is used against a MIM with a known mauling advantage, say δ. Thus, ε can be chosen as a function of δ. The synchronizing non-malleability of our composed scheme follows from the observation (made implicitly in [GPR16]), that the synchronizing non-malleability of the basic GPR scheme holds even when it is run in parallel with a (malleable) commitment scheme as long as the additional scheme satisfies an enhanced form of hiding security, known as *once rewindable hiding*. Fortunately, we show that our construction satisfies this extra property. While this is our overall strategy, we run into a technical issue related to our exact definition of ϵ-extractable commitments which does not seem sufficient to obtain the standard definition of non-malleability w.r.t. commitment. We resolve this issue by relying on a careful combinatorial analysis presented in Section VII.

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 synchro-

nizing 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.

II. 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,

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

A. Non-Malleable Commitments, and, ϵ-Extractable Commitments

In this section we define commitment schemes (Definition 1), non-malleable commitment schemes (Definition 3), and ϵ-extractable commitment schemes (Definition 4), the latter 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 $(m, w), (m', w')$ such that $m \neq m'$, at most one of $\mathsf{R}(c, m, w)$ and $\mathsf{R}(c, m', w')$ is 1;
- **Hiding:** For all m_0, m_1, $\big\{\mathrm{Com}(m_0; r)\big\}_r \approx_c \big\{\mathrm{Com}(m_1; r)\big\}_r$.

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].

Definition 2 (Rewind Secure Hiding). *We say that a three round commitment scheme $\langle \mathrm{C}, \mathrm{R}\rangle$ has* rewind secure hiding *if no PPT R^* can win the following game with probability noticeably better than $1/2$. In the following we let $(\mathrm{Com}_1, \mathrm{Com}_2, \mathrm{Com}_3)$ be the subroutines used to generate the messages in the three rounds.*

- R^* sends C two messages (m_0, m_1);
- C draws $b \leftarrow \{0,1\}$ and $\sigma_1 \leftarrow \mathrm{Com}_1(m_b)$ and sends σ_1 to R;
- R^* sends $\sigma_2, \hat{\sigma}_2$ to C;
- C prepares $\sigma_3 = \mathrm{Com}_3(\sigma_1, \sigma_2, m_b)$ and $\hat{\sigma}_3 = \mathrm{Com}_3(\sigma_1, \hat{\sigma}_2, m_b)$ and sends $(\sigma_3, \hat{\sigma}_3)$ to R^*;
- R^* outputs $b' \in \{0,1\}$ and wins if $b' = b$.

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 3 (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:*

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

Definition 4 (ϵ-Extractable Commitment). *We say that a perfectly binding commitment scheme $\langle \mathrm{C}, \mathrm{R}\rangle$ is* ϵ-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^*, $\Delta\big(\mathsf{val}^{\mathrm{C}^*}, \mathsf{Ext}_\varepsilon^{\mathrm{C}^*}\big) \leq \varepsilon$.*

B. Delayed-Input Rewind Secure Witness Indistinguishable Proofs

Definition 5 (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 6 (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 5) 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 III-B, 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 7 (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 5) 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^*, it holds that $\{\mathsf{REAL}^{V^*}(1^\lambda)\}_\lambda$ and $\{\mathsf{IDEAL}^{V^*}(1^\lambda)\}_\lambda$ are computationally indistinguishable, where the random variable $\mathsf{REAL}^{V^*}(1^\lambda)$ 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 his algorithm in the third round.*

Experiment $\mathsf{REAL}^{V^*}(1^\lambda)$: is defined as follows:

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^*.*

Experiment $\mathsf{IDEAL}^{V^*}(1^\lambda)$: is the output of the experiment $\mathsf{Sim}^{V^*}(1^\lambda, x; r)$.

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 7.*

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

C. 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 8. *Given $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \{0,1\}^\lambda$, let*

$\mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ *be the set*

$$\left\{\mathbf{r}_{S,i} \in \{0,1\}^\lambda : \emptyset \neq S \subset \{1,\ldots,k\}, i \in \{1,\ldots,\lambda\}\right\}.$$

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.

Definition 9. *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:*

1) **Valid Output:** $\Pr_{\mathbf{r} \leftarrow \{0,1\}^\lambda}\left[\mathsf{Pred}(\mathbf{r}) \neq \perp\right] \geq \varepsilon$;
2) **Prediction Advantage:** $\left|\Pr_{\mathbf{r} \leftarrow \{0,1\}^\lambda}\left[\mathsf{Pred}(\mathbf{r}) = \langle \mathbf{r}, \mathbf{x}\rangle \middle| \mathsf{Pred}(\mathbf{r}) \neq \perp\right] - \frac{1}{2}\right| \geq \varepsilon$.

Remark.: Prediction advantage says that, conditioned on Pred giving valid (*i.e.*, non$-\perp$) output, $\mathsf{Pred}(\mathbf{r}) = \langle \mathbf{r}, \mathbf{x}\rangle$ occurs with probability bounded away from $1/2$; the probability is over $\mathbf{r} \leftarrow \{0,1\}^\lambda$.

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 $\mathsf{T}_{\mathsf{Inv}} = \mathsf{poly}\big(\lambda, 1/\varepsilon, \mathsf{T}_{\mathsf{Pred}}\big)$, where $\mathsf{T}_{\mathsf{Pred}}$ is the running time of* Pred.

Definition 10. *We call the algorithm* Inv *guaranteed by Lemma 1 the* GL-inversion algorithm corresponding to Pred.

III. Building Blocks

A. 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) C $\to$ R**:** send $c = \mathrm{Com}(s, N; \eta)$ where $s, N \leftarrow \left[2^\lambda\right]$ and $\eta \leftarrow \$$;
2) R $\to$ C**:** draw and send $s' \leftarrow \left[2^\lambda\right]$;
3) **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

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

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 overwhelming probability $1 - 2/N$. The proof of the next claim is omitted because of space.

Claim 1. *Suppose one-to-one one-way functions exist. Let* R *be a polynomial time receiver such that $\Pr_{(s,N',\eta)}\left[\text{R completes protocol on receiving } c = \mathrm{Com}(s, N'; \eta)\right] \geq \varepsilon$, for non-negligible $\varepsilon > 0$, where the probability is over $s, N' \leftarrow \left[2^\lambda\right]$ and $\eta \leftarrow \$$. Then for any fixed polynomial $N = N(\lambda)$ there exists a polynomial time simulator* SIM *whose output is indistinguishable from the transcript resulting the execution of the above protocol with an honest* C*. Moreover, whenever* SIM *outputs the transcript of a completed protocol, the committed value in the transcript is N.*

B. Rewind Secure Delayed-Input Witness-Indistinguishable Proof

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.*

We omit the construction from this extended abstract. Please see the full version for details.

IV. Three Round ϵ-Extractible Bit-Commitment Scheme

A. The Scheme

Our scheme is described in Figure 1. Informally, our protocol consists of the following parts:

1) an unbounded polynomial commitment which C uses to commit to the value N;
2) a three round coin-flip type protocol:
   - C commits to random strings $(\mathbf{r}_1, \ldots, \mathbf{r}_\lambda)$ where $\mathbf{r}_i \in \{0,1\}^\lambda$;
   - R sends $\mathbf{r}'$
   - C sends a random $\mathbf{r} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ where $k = \mathcal{O}\big(\log(N)\big)$; we think of the string $\mathbf{r} \oplus \mathbf{r}'$ as being the "output" of this subroutine;
3) an interactive version of non-interactive commitment using output of the coin-flip protocol:
   - C sends $(f, \mathbf{y})$ a one-to-one one-way function and a random image $\mathbf{y} = f(\mathbf{x})$;
   - C sends $\langle \mathbf{r} \oplus \mathbf{r}', \mathbf{x}\rangle \oplus m$, where $m \in \{0,1\}$ is C's commitment bit;
4) C proves that $\mathbf{r} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ where $k = \mathcal{O}\big(\log(N)\big)$ and N is committed value inside the unbounded polynomial commitment.

As mentioned in the introduction, our proof of hiding follows the Goldreich-Levin based proof of hiding for the non-interactive commitment scheme. This proof uses an adversary who wins the hiding game in order to invert the one-way function. It is key for this proof that k be larger than roughly $\log\big(1/\varepsilon\big)$ where ε is the adversary's

advantage in the hiding game. If the committer is honest, $N = 1$ with high probability which means $k = 1$ as well. The unbounded polynomial commitment is used in order to argue that the real world is indistinguishable from a world where N is such that k is sufficiently large to allow the proof to go through.

We mention here that the above outline is an oversimplification of our actual protocol for two reasons. First, moving from the real world to the world where N is large requires (among other things) witness indistinguishability of the proof in part 4. In order to get the security proof to go through, we use two copies of steps 1 and 2, corresponding to witnesses w_0 and w_1. The second oversimplification, is that we require our WI proof to be have $1-$rewind secure delayed-input security. This is due to the fact that the Goldreich-Levin proof requires pairwise independence among many pairs of elements in $\mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_k)$. As C commits to the $\mathbf{r}_i$ in the first round of our protocol, we can only hope for a "computational analogue" of this pairwise independence to hold. We are able to establish the property if we use the scheme from Section III-B.

Theorem 4. *Assume that injective one-way functions exist. Then $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$ is a three-round, perfectly binding bit-commitment scheme.*

Note that perfect binding follows immediately from the perfect binding of Com. $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$ is also ϵ-extractable as per Definition 4. We omit the full proof for ϵ-extractablility since we give a full proof directly for our string commitment scheme in Section VI. However we give a sketch of the proof in the following.

Suppose an execution of $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$ is played with possibly malicious C^*, producing a transcript $\mathbb{T} = (\sigma_1, \sigma_2, \sigma_3)$. If the WI proof is non-accepting, the extractor simply outputs $\perp$. Else let the committed bit be m (observe that every transcript with an accepting WI has a valid committed bit). Suppose furthermore that C^*, $\mathbb{T}$ and σ_1 are such that the following two properties hold for some non-negligible $\varepsilon > 0$ (if not, we discuss what to do later):

1) if C^* is rewound and fed with a fresh $\hat{\sigma}_2 = \left((\hat{s}_0, \hat{s}_1), \hat{\mathbf{r}}', \hat{\mathsf{rwi}}_2\right)$, the probability that C^* returns a valid $\hat{\sigma}_3 = (\hat{\mathbf{r}}, \hat{v}, \hat{\mathsf{rwi}}_3)$ such that $\hat{\mathbf{r}} = \mathbf{r}$ is at least ε (by valid, we mean that $(\mathsf{rwi}_1, \hat{\mathsf{rwi}}_2, \hat{\mathsf{rwi}}_3)$ accepts); and
2) There exists a fixed bit $\hat{m}$ s.t. conditioned on C^* returning a valid $\hat{\sigma}_3$ with $\hat{\mathbf{r}} = \mathbf{r}$, the resulting transcript $\hat{\mathbb{T}} = (\sigma_1, \hat{\sigma}_2, \hat{\sigma}_3)$ is a commitment to $\hat{m}$ with probability at least $1/2 + \varepsilon$.

In this case, C^* and $\mathbb{T}$ can be used in a straightforward fashion to build a GL-predictor Pred, with advantage ε (see Definition 9). Then by Lemma 1, there is a GL-inverter which can recover $\mathbf{x}$ which allows extracting the committed bit m inside $\mathbb{T}$: $m = v \oplus \langle \mathbf{r} \oplus \mathbf{r}', \mathbf{x} \rangle$. The prediction algorithm takes input $\hat{\mathbf{r}}' \in \{0,1\}^\lambda$, completes $\hat{\mathbf{r}}'$ to $\hat{\sigma}_2 = \left((\hat{s}_0, \hat{s}_1), \hat{\mathbf{r}}', \hat{\mathsf{rwi}}_2\right)$ by choosing fresh $\hat{s}_0, \hat{s}_1, \hat{\mathsf{rwi}}_2$ and sends $\hat{\sigma}_2$ to C^*. If C^* returns $\hat{\sigma}_3 = (\hat{\mathbf{r}}, \hat{v}, \hat{\mathsf{rwi}}_3)$ such that $(\mathsf{rwi}_1, \hat{\mathsf{rwi}}_2, \hat{\mathsf{rwi}}_3)$ is an accepting proof and such that $\hat{\mathbf{r}} = \mathbf{r}$, Pred outputs $\hat{v} \oplus v$; otherwise Pred outputs $\perp$. Property 1 above establishes that Pred meets the "valid output" requirement of being a GL-algorithm, while property 2 establishes "prediction advantage". Indeed, conditioned on C^* returning valid $\hat{\sigma}_3$ with $\hat{\mathbf{r}} = \mathbf{r}$, with probability at least

Parameters and Subroutines: The protocol takes place between C and R. Let λ be the security parameter. Let $\mathcal{F}_{1-1\ \mathsf{owf}}$ be a family of one-to-one oneway functions taking inputs of length λ. Let Com be a non-interactive, perfectly binding commitment scheme. Let $(\mathsf{RWI}_1, \mathsf{RWI}_2, \mathsf{RWI}_3, \mathsf{RWI}_4)$ be the three-round $B-$rewind secure delayed-input WI proof (RWI_4 is verification).

Input: C has an input bit $m \in \{0,1\}$ that it will commit to; R uses no input.

Commit Phase:

1. $\mathrm{C} \longrightarrow \mathrm{R}$**:** C sends $\left((c_0, c_1); \{(z_0^\alpha, z_1^\alpha)\}_{\alpha=1,\ldots,\lambda}; f, \mathbf{y}; \mathsf{rwi}_1\right)$ to R where:
(a) $s_0, s_1, N_0, N_1 \leftarrow \left[2^\lambda\right]$; $\eta_0, \eta_1 \leftarrow \$$, $c_a = \mathrm{Com}(s_a \circ N_a; \eta_a)$ for $a = 0, 1$;
(b) $\mathbf{r}_a^\alpha \leftarrow \{0,1\}^\lambda$, $\omega_a^\alpha \leftarrow \$$, $z_a^\alpha = \mathrm{Com}(\mathbf{r}_a^\alpha; \omega_\alpha^a)$ for $a = 0, 1$, $\alpha = 1, \ldots, \lambda$;
(c) $\mathsf{rwi}_1 \leftarrow \mathsf{RWI}_1$; the statement will come during the third round;
(d) $f \leftarrow \mathcal{F}_{1-1\ \mathsf{owf}}$, $\mathbf{x} \leftarrow \{0,1\}^\lambda$, $\mathbf{y} = f(\mathbf{x})$;

2. $\mathrm{R} \longrightarrow \mathrm{C}$**:** R sends $\left((s_0', s_1'); \mathbf{r}'; \mathsf{rwi}_2\right)$ to C where:
(a) $s_0', s_1' \leftarrow \left[2^\lambda\right]$; (b) $\mathbf{r}' \leftarrow \{0,1\}^\lambda$; (c) $\mathsf{rwi}_2 \leftarrow \mathsf{RWI}_2$.

3. $\mathrm{C} \longrightarrow \mathrm{R}$**:** C sends $\left(\mathbf{r}; v; \mathsf{rwi}_3\right)$ to R where:
(a) $b \leftarrow \{0,1\}$, $\mathbf{r} = \mathbf{r}_b^1$; (b) $v = \langle \mathbf{x}, \mathbf{r} \oplus \mathbf{r}' \rangle \oplus m$;
(c) rwi_3 generated using RWI_3 and $(\mathsf{rwi}_1, \mathsf{rwi}_2)$ for statement: $\exists\ \left(b, s, N, \eta, \mathsf{val}, \{\mathbf{r}_\alpha, \omega_\alpha\}_{\alpha \in [\mathsf{val}]}\right)$ st

$$(i)\ \mathsf{val} = \begin{cases} 3\log(\lambda) + 9\log(N_b), \\ s + s_b' \equiv 0 \pmod{N_b} \\ 1, \\ \text{otherwise} \end{cases}$$

(ii) $c_b = \mathrm{Com}(s \circ N; \eta)$
(iii) $z_b^\alpha = \mathrm{Com}(\mathbf{r}_\alpha; \omega_\alpha)\ \forall\ \alpha \in [\mathsf{val}]$
(iv) $\mathbf{r} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_{\mathsf{val}})$

Decommit Phase: C sends $\mathbf{x}$ to R along with decommitments of the first round commitments c_a and z_a^α for $\alpha = 1, \ldots, \lambda$ and $a = 0, 1$.

Output: If the verification in RWI_4 accepts, then R outputs $v \oplus \langle \mathbf{r} \oplus \mathbf{r}', \mathbf{x} \rangle$.

Figure 1. Three-Round ϵ-Extractable Bit Commitment Scheme $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$

$1/2+\varepsilon$,

$$\begin{aligned}\hat{v}\oplus v &= \big(\hat{m}\oplus\langle\hat{\mathbf{r}}\oplus\hat{\mathbf{r}}',\mathbf{x}\rangle\big)\oplus\big(m\oplus\langle\mathbf{r}\oplus\mathbf{r}',\mathbf{x}\rangle\big)\\ &= \langle\hat{\mathbf{r}}',\mathbf{x}\rangle\oplus\Big(\langle\mathbf{r}',\mathbf{x}\rangle\oplus m\oplus\hat{m}\Big).\end{aligned}$$

So depending on whether $\big(\langle\mathbf{r}',\mathbf{x}\rangle\oplus m\oplus\hat{m}\big)$ is 0 or 1, the output of Pred either is equal or not equal to $\langle\hat{\mathbf{r}}',\mathbf{x}\rangle$ with probability $1/2+\varepsilon$.

If condition 1 or 2 is not true for the chosen $\mathbf{r}$, simply try again with a fresh choice of the first transcript with the same first message σ_1. If condition 1 is not true for any choice of $\mathbf{r}$, simply output $\perp$. If condition 2 is not true for any choice of $\mathbf{r}$, then simply output a random bit (since in this case C^* is committing to a random bit for σ_1). Note that ε is given as input to the extractor and it can check these conditions by running in time poly$(1/\varepsilon)$.

V. Hiding Overview

Lemma 2. *Assume that one-to-one one-way functions exist. Then $\langle\mathrm{C},\mathrm{R}\rangle_{\mathsf{ext-bit}}$ is computationally hiding.*

Let $\mathsf{G}_0^{\mathcal{A}}(m)$ denote the real world experiment where a challenger $\mathcal{C}$ sends an adversarial receiver $\mathcal{A}$ a commitment to $m\in\{0,1\}$. The transcript of $\mathsf{G}_0^{\mathcal{A}}(m)$ is:

$$\big((c_0,c_1),(s_0',s_1');\{(z_0^\alpha,z_1^\alpha)\}_{\alpha=1,\ldots,\lambda},\mathbf{r}',\mathbf{r};f,\mathbf{y},v;(\mathsf{rwi}_1,\mathsf{rwi}_2,\mathsf{rwi}_3)\big)$$

where (among other things) $\mathbf{y}=f(\mathbf{x})$ for a randomly drawn $\mathbf{x}\leftarrow\{0,1\}^\lambda$, and $v=\langle\mathbf{r}\oplus\mathbf{r}',\mathbf{x}\rangle\oplus m$. We prove hiding by showing that for all polynomial time $\mathcal{A}$, $\mathsf{G}_0^{\mathcal{A}}(0)\approx_{\mathsf{c}}\mathsf{G}_0^{\mathcal{A}}(1)$. The omitted proof consists of three main steps. First, we define a hybrid game $\mathsf{G}_1^{\mathcal{A},\delta}(m)$ (parametrized by $\delta>0$) and prove that for all non-negligible $\delta>0$, and $m\in\{0,1\}$, $\mathsf{G}_0^{\mathcal{A}}(m)\approx_{\mathsf{c}}\mathsf{G}_1^{\mathcal{A},\delta}(m)$. Next, we use a distinguisher for $\mathsf{G}_1^{\mathcal{A},\delta}(0)$ and $\mathsf{G}_1^{\mathcal{A},\delta}(1)$ for a particular $\delta>0$ (related to $\mathcal{A}$'s chance of breaking hiding), to construct a prediction algorithm Pred which, given $\big(f,f(\mathbf{x})\big)$, predicts random inner products of $\mathbf{x}$. Finally, we use this prediction algorithm to construct an inversion algorithm Inv which, given $\big(f,f(\mathbf{x})\big)$, outputs $\mathbf{x}$ with non-negligible probability (over f and $\mathbf{x}$). This contradicts the assumption that $\mathcal{F}_{\mathsf{1-1\ owf}}$ is a family of one-way functions, proving $\mathsf{G}_1^{\mathcal{A},\delta}(0)\approx_{\mathsf{c}}\mathsf{G}_1^{\mathcal{A},\delta}(1)$, from which hiding follows.

Our proof of hiding follows the high-level strategy of the proof of the Goldreich-Levin theorem [GL89]. However several complications arise, essentially due to the fact that our protocol is interactive. For example, since C commits in the first round to its set of potential third round messages, we will not have the freedom to run our prediction algorithm on arbitrary strings as is normally done in Goldreich-Levin-type proofs. Instead, we have to use cryptographic arguments to show that the set of strings which are allowed to be given to the prediction algorithm are "random enough" against polynomial time adversaries that the proof goes through anyway. The reader familiar with Rackoff's proof of the Goldreich-Levin theorem recalls that pairwise-independence played a key role. Indeed, the most technical part of our proof is the establishment of a "pairwise-independence-like" property.

VI. Three Round ϵ-Extractable String Commitment Scheme

In this section we describe a three round ϵ-extractable commitment scheme from any one-to-one one-way function. Intuitively, in order to commit to a string $m\in\{0,1\}^e$, let $(m_1,\ldots,m_n)=\mathsf{ECC}(m)\in\{0,1\}^n$ for ECC a binary error correcting code with constant distance, then use $\langle\mathrm{C},\mathrm{R}\rangle_{\mathsf{ext-bit}}$ to commit to each m_i in parallel. However, there are a few complications which we discuss momentarily. The formal description of $\langle\mathrm{C},\mathrm{R}\rangle$ is given in Figure 2.

Intuitively, the protocol can be broken into the following parts.

1) C sends $z=\mathrm{Com}(m;\omega)$, and computes $(c_1,\ldots,c_n)=\mathsf{ECC}(m\circ\omega)$.
2) C and R engage in n *simplified* executions of $\langle\mathrm{C},\mathrm{R}\rangle_{\mathsf{ext-bit}}$ where C commits to c_i in the $i-$th copy. The simplified protocols go as follows:
   - C sends $f,\mathbf{y},z$ where $z=\mathrm{Com}(\mathbf{r})$; R sends $\mathbf{r}'$; C sends $\mathbf{r}$ and $v=\langle\mathbf{x},\mathbf{r}\oplus\mathbf{r}'\rangle\oplus c_i$;

   The reason for the simplifications is that there will be a single global proof, unbounded polynomial commitment, and Goldreich-Levin set for the entire outer protocol, so the inner protocols do not need these parts.
3) C also commits to random strings $\mathbf{r}^1,\ldots,\mathbf{r}^\lambda$, which are generators to a Goldreich-Levin set. Also, C and R play an unbounded polynomial commitment to determine the size of the GL set.
4) Finally, C proves that every value $\mathbf{r}$ send in the inner, simplified $\langle\mathrm{C},\mathrm{R}\rangle_{\mathsf{ext-bit}}$ sessions, is the committed string inside the corresponding z, except for at most one such $\mathbf{r}$ which is in the GL set of size determined by N.

We remark again that the above is a simplification. The actual protocol employs bounded rewind secure WI, and so needs two copies of much of the above data for security.

Parameters and Subroutines.: The protocol takes place between a committer C and a receiver R. Let λ be the security parameter. Let $\mathcal{F}_{\mathsf{1-1\ owf}}$ be a family of one-to-one oneway functions taking inputs of length λ. Let Com be a non-interactive, perfectly binding (string) commitment scheme, whose decommitments are strings of length k. Let $\mathsf{ECC}:\{0,1\}^k\to\{0,1\}^n$ be a binary error-correcting code with constant distance. Finally, let RWI be the three-round rewind secure WI proof from Section III-B.

Theorem 5. *Assume that one-to-one one-way functions exist. Then $\langle\mathrm{C},\mathrm{R}\rangle$ is a three-round, perfectly binding, computationally hiding, $\varepsilon-$extractable commitment scheme.*

Proof Outline: Perfect binding follows immediately from the perfect binding of $\langle\mathrm{C},\mathrm{R}\rangle_{\mathsf{ext-bit}}$. The main challenge is proving $\varepsilon-$extractability as per Definition 4,

Input: C has a message $m \in \{0,1\}^e$ that it will commit to; R uses no input.

Commit Phase:

1. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $\big(z; (c_0, c_1); \{(z_0^\alpha, z_1^\alpha)\}_{\alpha\in[\lambda]}; \{f_i, \mathbf{y}_i, z_i^*\}_{i\in[n]}; \mathsf{rwi}_1\big)$ to R:

(a) $z = \mathrm{Com}(m;\omega)$ for $\omega \leftarrow \$$;
(b) $c_a = \mathrm{Com}(s_a \circ N_a; \eta_a)$ where $s_a, N_a \leftarrow \big[2^\lambda\big]$, $\eta_a \leftarrow \$$;
(c) $z_a^\alpha = \mathrm{Com}(\mathbf{r}_a^\alpha; \omega_a^\alpha)$ where $\mathbf{r}_a^\alpha \leftarrow \{0,1\}^\lambda$, $\omega_a^\alpha \leftarrow \$$;
(d) $f_i^a \leftarrow \mathcal{F}_{1-1\ \mathsf{owf}}$, $\mathbf{x}_i \leftarrow \{0,1\}^\lambda$, $\mathbf{y}_i = f_i(\mathbf{x}_i)$; $z_{i,a}^* = \mathrm{Com}(\mathbf{r}_i^*; \omega_i^*)$ for $\mathbf{r}_{i,a}^* \leftarrow \{0,1\}^\lambda$, $\omega_{i,a}^* \leftarrow \$$;
(e) $\mathsf{rwi}_1 \leftarrow \mathsf{RWI}_1$; the statement will come during the third round;

2. $\underline{\mathrm{R} \longrightarrow \mathrm{C}}$**:** R sends $\big((s_0', s_1'); \{\mathbf{r}_i'\}; \mathsf{rwi}_2\big)$ to C:

(a) $s_0', s_1' \leftarrow \big[2^\lambda\big]$; (b) $\mathbf{r}_i' \leftarrow \{0,1\}^\lambda$; (c) $\mathsf{rwi}_2 \leftarrow \mathsf{RWI}_2$.

3. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $\big(\{\mathbf{r}_i, v_i\}; \mathsf{rwi}_3\big)$ to R where:

(a) $\mathbf{r}_i = \mathbf{r}_{i,b}^*$; for $b \leftarrow \{0,1\}$ (b) $v_i = \langle \mathbf{x}_i, \mathbf{r}_i \oplus \mathbf{r}_i' \rangle \oplus c_i$, where $(c_1, \ldots, c_n) = \mathsf{ECC}(m \circ \omega)$;
(c) rwi_3 so that $(\mathsf{rwi}_1, \mathsf{rwi}_2, \mathsf{rwi}_3)$ proves: $\exists$ $\big(b, i^*, s, N, \eta, \mathsf{val}, \{\mathbf{r}_\alpha, \omega_\alpha\}_{\alpha\in[\mathsf{val}]}, \{\overline{\mathbf{r}}_i^*, \overline{\omega}_i^*\}_{i\in[n]}\big)$ st

(i) $\mathsf{val} = \begin{cases} 3\log(\lambda) + 9\log(N_b), & N_b | s + s_b' \\ 1, & \text{otherwise} \end{cases}$
(ii) $c_b = \mathrm{Com}(s \circ N; \eta)$
(iii) $z_b^\alpha = \mathrm{Com}(\mathbf{r}_\alpha; \omega_\alpha)\ \forall\ \alpha \in [\mathsf{val}]$
(iv) $\mathbf{r}_{i^*} \in \mathsf{GL}(\mathbf{r}_1, \ldots, \mathbf{r}_{\mathsf{val}}) \cup \{\overline{\mathbf{r}}_{i^*}^*\}$
(v) $z_i^* = \mathrm{Com}(\overline{\mathbf{r}}_i^*; \overline{\omega}_i^*)$
(vi) $\mathbf{r}_i = \overline{\mathbf{r}}_i^*\ \forall\ i \neq i^*$

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 decodes $(\hat{c}_1, \ldots, \hat{c}_n)$ to get (m, ω). If any errors are detected during decoding, or if $(m,\omega) \neq \mathrm{Decom}(z)$, output $\perp$. Otherwise, output m.

Figure 2. Three-Round ε-Extractable Commitment Scheme $\langle \mathrm{C}, \mathrm{R} \rangle$

our proof occupies the next several sections. Hiding is proved using a hybrid argument where we pass from a commitment to m to a commitment of m' by changing C's inputs in the $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$ from c_i to 0, one at a time. If we instantiate the $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-str}}$ with a version of $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-bit}}$ which has $1-$rewind hiding, then $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-str}}$ will have $1-$rewind hiding security as well. ■

VII. Three-Round NM Commitment from One-to-one OWF

Our three round non-malleable commitment is very simple. It is shown in Figure 3. It consists simply of two commitments run side by side. The first, GPR, is the main protocol from [GPR16]; the second is the $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-str}}$ from Section VI, enhanced to have rewindable hiding security.

Parameters and Subroutines: The protocol takes place between a committer C and a receiver R. Let $(\mathrm{Com}_1, \mathrm{Com}_2, \mathrm{Com}_3)$ denote the subroutines used to generate the rounds of $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-str}}$ from Section VI. Let $(\mathsf{GPR}_1, \mathsf{GPR}_2, \mathsf{GPR}_3)$ represent the subroutines of GPR, the main protocol from [GPR16] that is non-malleable against a synchronizing adversary.

Input: C has $m \in \{0,1\}^\lambda$ that it will commit to; R uses no input.

Commit Phase:

1. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $(\mathsf{gpr}_1, \mathrm{Com}_1)$ to R where:

(a) $\mathsf{gpr}_1 \leftarrow \mathsf{GPR}_1(m)$; let ω be the decommitment information of gpr_1.
(b) $\sigma_1 \leftarrow \mathrm{Com}_1(\omega)$.

2. $\underline{\mathrm{R} \longrightarrow \mathrm{C}}$**:** R sends $(\mathsf{gpr}_2, \mathrm{Com}_2)$ to C.

3. $\underline{\mathrm{C} \longrightarrow \mathrm{R}}$**:** C sends $(\mathsf{gpr}_3, \mathrm{Com}_3)$ to R.

Decommit Phase: C sends R the decommitment information corresponding $(\mathsf{gpr}_1, \mathsf{gpr}_2, \mathsf{gpr}_3)$ and $(\sigma_1, \sigma_2, \sigma_3)$.

Output: If both decommitments are well formed, and if, moreover, the committed message in $(\sigma_1, \sigma_2, \sigma_3)$ is the decommitment information for gpr_1, then R outputs m, the committed message in $(\mathsf{gpr}_1, \mathsf{gpr}_2, \mathsf{gpr}_3)$; otherwise R outputs $\perp$.

Figure 3. Three-Round Non-Malleable Commitment Scheme $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{nm}}$

Theorem 6. *If one-to-one oneway functions exist then $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{nm}}$ is a three-round, perfectly binding, non-malleable commitment scheme.*

Proof: Perfect binding and hiding follow immediately from the perfect binding and hiding of GPR and $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{ext-str}}$. To prove non-malleability, we consider the case when the MIM is synchronizing (*i.e.*, plays the corresponding messages from the two sessions one after another), or sequential (*i.e.*, plays the entire left interaction followed by the entire right). The synchronizing case follows from [GPR16], who proved that GPR is non-malleable against a synchronizing adversary *even if* it is run in parallel with a (malleable) commitment scheme which has $1-$rewind secure hiding. The main difficulty is to prove non-malleability against a sequential MIM; our proof occupies the next section. ■

A. Non-Malleability of $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{nm}}$ Against a Sequential MIM

Notation.: We are interested in the following randomized process, parametrized by $m \in \{0,1\}^\lambda$ and a PPT sequential MIM M: M plays two executions of the commit phase of $\langle \mathrm{C}, \mathrm{R} \rangle_{\mathsf{nm}}$, one on the left where M plays as receiver against an honest C, who commits to m, and

one on the right where M plays as committer against honest R. We $\mathbb{T}$ to denote the transcript of this experiment:

$$\mathbb{T} = (\mathbb{T}_{\mathsf{L}}, \mathbb{T}_{\mathsf{R}}) = \big(\mathbb{T}_{\mathsf{L}}, (\mathsf{gpr}_1, \mathsf{gpr}_2, \mathsf{gpr}_3), (\tau_1, \tau_2, \tau_3), (\sigma_1, \sigma_2, \sigma_3)\big),$$

where $(\mathsf{gpr}_1, \mathsf{gpr}_2, \mathsf{gpr}_3)$ is the GPR transcript of the right execution, and $\big((\tau_1, \tau_2, \tau_3), (\sigma_1, \sigma_2, \sigma_3)\big)$, is the right $\langle \mathrm{C}, \mathrm{R}\rangle_{\mathsf{ext-str}}$ transcript, notated as in Section VI. Proving non-malleability ammounts to showing that that for every non-negligible $\varepsilon > 0$, polytime MIM M, there exists a simulator $\mathsf{SIM}_{\varepsilon}^{\mathsf{M}}$ which outputs a transcript/message pair $(\mathbb{T}, \tilde{m})$ such that for all $m \in \{0,1\}^{\lambda}$, and polytime distinguishers D:

$$\Big|\Pr_{(\mathbb{T},\tilde{m})\leftarrow \mathsf{MIM}_m^{\mathsf{M}}}\big[\mathsf{D}(\mathbb{T},\tilde{m}) = 1\big] - \Pr_{(\mathbb{T},\tilde{m})\leftarrow \mathsf{SIM}_{\varepsilon}^{\mathsf{M}}}\big[\mathsf{D}(\mathbb{T},\tilde{m}) = 1\big]\Big| \leq \varepsilon, \tag{2}$$

where recall that $\mathsf{MIM}_m^{\mathsf{M}}$ plays the MIM experiment with M, producing a transcript $\mathbb{T}$ and then outputs $(\mathbb{T}, \tilde{m})$ where $\tilde{m} = \mathsf{val}(\mathbb{T})$ is the committed value inside the right execution. The main Lemma of this section is thus:

Lemma 3 (Sequential NM of $\langle \mathrm{C}, \mathrm{R}\rangle_{\mathsf{nm}}$). *For all non-negligible $\varepsilon > 0$ and polynomial time sequential M, there exists a polynomial time algorithm $\mathsf{SIM}_{\varepsilon}^{\mathsf{M}}$ which runs in time $\mathsf{poly}\big(\lambda, \mathsf{T}_{\mathsf{M}}, 1/\varepsilon\big)$, and on input id, outputs a transcript-message pair $(\mathbb{T}, \tilde{m})$ such that for all $m, \mathsf{id} \in \{0,1\}^{\lambda}$, and polynomial time distinguishers D, (2) holds.*

Proof Intuition.: The key to the proof is an extraction procedure, $\mathsf{EXTLOOP}$, which yields several distinct candidate simulators, one of which is the simulator we are looking for. $\mathsf{EXTLOOP}$ is an extraction loop, the $k-$th candidate simulator, denoted SIM_k, works by executing the loop k times and then exiting and giving output. On the $i-$th run through the loop, the simulation procedure will call D_i, a PPT distinguisher for SIM_{i-1} (if no such D_i exists then SIM_{i-1} is the simulator we are looking for, so we are done). Each time through the extraction loop, we are likely to recover a oneway function preimage from the right execution of $\langle \mathrm{C}, \mathrm{R}\rangle_{\mathsf{ext-str}}$. The lemma follows since if we get every oneway function preimage, we can recover M's commitment in polynomial time and thus he cannot be mauling (this would break hiding).

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.

[CGJ18] Arka Rai Choudhuri, Vipul Goyal, and Abhishek Jain. Round optimal secure multiparty computation from almost minimal assumptions. In *Under submisison to this conference (STOC 2019)*, 2018.

[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.