Two-Round Maliciously Secure Computation with Super-Polynomial Simulation

Amit Agarwal* James Bartusek** Vipul Goyal***
Dakshita Khurana† Giulio Malavolta‡

Abstract. We propose the first maliciously secure multi-party computation (MPC) protocol for general functionalities in two rounds, without any trusted setup. Since polynomial-time simulation is impossible in two rounds, we achieve the relaxed notion of superpolynomial-time simulation security [Pass, EUROCRYPT 2003]. Prior to our work, no such maliciously secure protocols were known even in the two-party setting for functionalities where both parties receive outputs. Our protocol is based on the sub-exponential security of standard assumptions plus a special type of non-interactive non-malleable commitment.
At the heart of our approach is a two-round multi-party conditional disclosure of secrets (MCDS) protocol in the plain model from bilinear maps, which is constructed from techniques introduced in [Benhamouda and Lin, TCC 2020].

1 Introduction

A multi-party computation (MPC) protocol [GMW87] allows a set of n mutually distrustful parties to securely compute any function f on their inputs $(x_1, \ldots, x_n)$, while revealing nothing beyond the function output $f(x_1, \ldots, x_n)$. An MPC satisfies the notion of *semi-honest* security if the privacy of the inputs is guaranteed against an adversary that faithfully follows the specification of the protocols. On the other hand, if the MPC is secure against *any* adversary, who can corrupt any subset of parties and let them deviate from the protocol specifications arbitrarily, then we say that it satisfies the notion of *malicious* security.

MPC is a central tool in modern cryptography and characterizing its exact round complexity has been a major open problem. Recently, this question was settled for the semi-honest setting [GS18,BL18a] where the authors showed a "round-collapsing" compiler to turn any MPC protocol into a 2-round protocol, under the (minimal) assumption of the existence of a 2-round oblivious transfer (OT) protocol. Unfortunately, the compiled protocols achieve only semi-honest

* UIUC. amita2@illinois.edu
** UC Berkeley. bartusek.james@gmail.com
*** NTT Research and CMU. vipul@cmu.edu
† UIUC. dakshita@illinois.edu
‡ Max Planck Institute for Security and Privacy. giulio.malavolta@hotmail.it

security (even if the input protocols were maliciously secure to begin with). Achieving malicious security requires one to add additional rounds of interaction [BHP17,ACJ17,HHPV18,BL18a,BGJ+18,CCG+19] or assume the presence of a trusted setup [GS18]. Besides introducing an additional (reusable) round of interaction where all participants need to receive the common reference string (CRS), the presence of a trusted setup is at odds with the main objective of MPC of reducing the trust in external parties. This motivates us to ask the following question:

Can we construct maliciously secure 2-round MPC without trusted setup?

At first, it might appear that the answer to the above question is clearly negative: Even for the 2-party setting it is well known that four rounds are necessary [KO04] (with respect to blackbox simulation) and that polynomial time simulation in 2 rounds is strictly impossible [GO94]. However none of these barriers hold if we consider the relaxed notion of superpolynomial-time simulation.

Super-Polynomial Simulation (SPS). SPS-based security [Pas03,PS04] has emerged as the de-facto notion of security to bypass impossibility results of classical polynomial-time simulation. In SPS security, the adversary is restricted to run in (non-uniform) polynomial time but the simulator is allowed to run in super-polynomial time. To see why this is a meaningful notion, note that the standard definition of input-indistinguishability (e.g. semantic security for the case of encryption) is equivalent to SPS security with an *unbounded* simulator. Thus, input-indistinguishability is a strict relaxation of SPS security.

In fact, the notion of (malicious) 2-round MPC with SPS security has been recently considered for the restricted settings of 2 parties (2PC), out of which one might be corrupted. Recent works [BGI+17,JKKR17,MPP20] achieve 2-round 2PC with SPS security from a variety of assumptions, where a single party receives the output. Even constructing a 2-round 2PC with SPS security where both parties receive the output at the end of second round is currently an open problem,[1] let alone extending such a result to the setting of more than 2 parties.

As discussed in [BGJ+17], who construct *three round* MPC with SPS security, it is helpful to view SPS security through the lens of the security loss inherent in all security reductions. In polynomial-time simulation, the security reduction has a polynomial security loss with respect to the ideal world. That is, an adversary in the real world has as much power as another adversary that runs in polynomially more time in the ideal world. In SPS security, the security reduction has a fixed *super-polynomial* security loss, for example 2^{n^ϵ} for a small constant $\epsilon > 0$ and security parameter n, with respect to the ideal world. Just as in other applications in cryptography using super-polynomial assumptions, this situation still guarantees security as long as the ideal model is itself super-polynomially secure. For instance, if the ideal model hides honest party inputs

[1] Running two instances of the same protocol in parallel does not achieve any meaningful security guarantee since nothing prevents one party from using two different inputs in each session.

information-theoretically, then security is maintained even with SPS. This is true for applications like online auctions, where no information is leaked in the ideal world about honest party inputs beyond what can be easily computed from the output. But SPS also guarantees security for ideal worlds with cryptographic outputs, like blind signatures, as long as the security of the cryptographic output is guaranteed against super-polynomial adversaries. Indeed, SPS security was explicitly considered for blind signatures in [GRS$^+$11,GG14] with practically relevant security parameters computed in [GG14].

1.1 Our Results

We construct a 2-round MPC protocol for polynomially-many parties with SPS security. All communications happen via a broadcast channel that immediately relays the messages to all participants. We guarantee security in the *dishonest majority* setting and against malicious adversaries, i.e. we allow the adversary to behave arbitrarily and to corrupt all but one participant. We do not assume a trusted setup or a common reference string (i.e. our protocol is in the plain model). [2] More concretely, we obtain the following result.

Theorem 1 (Informal). *Assuming the sub-exponential security of the following building blocks:*

- *A non-interactive witness-indistinguishable (NIWI) proof.*
- *A special non-interactive non-malleable commitment scheme*[3]*.*
- *A 2-round semi-malicious*[4] *MPC.*
- *A bilinear group in which the SXDH assumption holds.*

Then there exists a 2-round MPC in the plain model with SPS security for all functions.

Prior to our work, 2-round (malicious) MPC was known only for the 2-party settings where only one party receives the output at the end of the interaction [BGI$^+$17,JKKR17,MPP20]. Protocols for more than 2 parties in the plain model were not known under any assumption. Note that 3-round MPC with SPS (and even concurrent) security is known (see [BGJ$^+$17] and references therein) and that 1-round MPC is impossible in the plain model (even with SPS-security). Thus, our work fills the natural knowledge gap about the round complexity of MPC with SPS security.

[2] We note that our usage of bilinear group based NIWI does not require any setup phase as the prover can self sample the group. Soundness of NIWI will hold as long as the group is cyclic and of the right order[GOS06b]. Also, our usage of tag-based non-malleable commitment scheme doesn't require setup as the parties can locally choose their identities.

[3] Specifically, we assume (a strengthened form of) sub-exponentially secure non-malleable commitments with respect to commitment.

[4] Semi-malicious security is a strengthening of semi-honest security where the adversary follows the specifications of the protocols but can choose the random coins of the corrupted parties arbitrarily.

Multi-Party Conditional Disclosure of Secrets. The central tool that we use to achieve our main result is a new construction of multi-party conditional disclosure of secrets (MCDS). Loosely speaking, in an MCDS protocol we want one party (the sender) to reveal a message to a set of n parties (the receivers) if and only if some statements $(x_1, \ldots, x_n)$ are all true. Each receiver holds a witness w_i and, at the end of the interaction, the message m can be *publicly reconstructed* if all witnesses are valid, i.e. $(w_i, x_i) \in \mathcal{R}$, where $\mathcal{R}$ is an NP relation. Security requires that all witnesses remain hidden and that the message m is also hidden if at least one statement x_i is false. Building on the recent techniques of [BL20], we obtain the following construction, which may be of independent interest.

Theorem 2 (Informal). *If there exists a bilinear group where the SXDH and the DLin problems are (subexponentially) hard, then there exists a (delayed statement) 2-round (subexponentially-secure) MCDS protocol for NP in the plain model.*

On the Assumptions. We observe that all our building blocks except non-interactive non-malleable commitment admit efficient instantiations from standard assumptions over bilinear maps. In the case of constant number of parties, we achieve the required special non-malleable commitments by relying on the RSW time-lock puzzle assumption in [LPS17] together with any sub-exponentially quantum-hard non-interactive commitment (which follows, e.g., from quantum-hardness of LWE). In the case of polynomially many parties, our special non-malleable commitments can be instantiated based on a variant of a "hardness amplifiability" assumption on non-interactive commitments (inspired by [BL18b]), together with other standard assumptions. A much simpler instantiation of the required non-malleable commitments for polynomially many parties would also follow from the factoring-based adaptive one-way functions of [PPV08] together with any sub-exponentially quantum-hard non-interactive commitment (which follows, e.g., from quantum sub-exponential hardness of LWE).

Conclusion and Open Questions. This work provides the first template to achieve multi-party computation in two rounds against Byzantine adversaries without trusted setup. Prior to our work, all existing two round multi-party (and even two-party) computation protocols [GGHR14,GP15,GS18,BL18a,BJKL21] either required trusted setup or achieved provable security only against variants of honest-but-curious adversaries. On the other hand, our protocol achieves security with super-polynomial simulation against arbitrary malicious corruptions.

We believe that future work will be able to build on this template to realize secure two-round MPC protocols under a variety of different assumptions. For instance, improved constructions of non-interactive non-malleable commitments that rely on various new "axes of hardness" could improve the assumptions used in our work. The problem of building multi-party CDS under other standard assumptions could also be an interesting open question for future work.

1.2 Technical Overview

We first describe our principal building block - a construction of multi-party conditional disclosure of secrets (MCDS) in the plain model - and then describe the techniques we develop to construct two-round MPC in the plain model.

Multi-Party Conditional Disclosure of Secrets As discussed in the previous section, our two-round maliciously secure MPC protocol relies on an underlying semi-malicious MPC protocol. The first challenge that we encounter in compiling this to a maliciously secure MPC is the following: there needs to be a mechanism to make sure that the first message of each party is well-formed, otherwise the semi-malicious MPC offers no security whatsoever. Now in the absence of a trusted setup, we cannot simply attach a NIZK proof that certifies well-formedness. This forces us to adopt an *implicit* approach instead.

Specifically, instead of relying on publicly-verifiable NIZKs, we aim to realize the following (two-round) *two*-party functionality: Let C be an NP-verification circuit that the parties wish to compute over some secret witness w. One party - the receiver - has a witness w as input, the other party - the sender - has a secret message m as input. The public output is m if $C(w) = 1$, and otherwise the output is $\perp$.

This functionality would allow us to achieve the desired goal, since we can "condition" the transfer of the second round message to the fact that the first round message of all parties was well-formed. In the multi-party settings, all parties should simultaneously receive all second round messages, and therefore we additionally need to ensure that the above functionality satisfies *public reconstruction*: If $C(w) = 1$, then the message m is publicly recoverable from the conversation transcript. While this appears to be a plausible avenue to attack the problem, building a protocol implementing this functionality in two rounds and in the plain model requires some new ideas. We note that the notion of CDS and its use as an alternative to zero-knowledge was first introduced in the work of [GIKM98].

What Makes This a Difficult Problem? Since parties may behave maliciously, there is no guarantee that a party A's first round message is honestly generated. Furthermore, A should be able to recover an output after obtaining party B's second round message, which is computed based on A's potentially mal-formed first message. Thus, it appears that B should have some guarantee that A's first message is well-formed before it computes and releases its second round message, which will potentially reveal information about its secret input. Importantly, this proof of well-formedness should preserve the confidentiality of A's input.

In the CRS model, one could have each party prove the well-formedness of its first round message with a NIZK. However, in the absence of any setup, one cannot achieve such strong zero-knowledge properties with a non-interactive proof. The best we can hope for is to have each party prove that its first message is well-formed with a non-interactive witness indistinguishable proof (NIWI). Now, in order to preserve confidentiality while using a NIWI, there must exist multiple valid explanations (i.e. witnesses) of the party's first round message. Thus, a natural approach is to have each party generate two separate first round messages and prove with a NIWI that at least one of the two is well-formed.

While this appears promising, there are still serious issues that prevent one from constructing general-purpose two-round two-party computation in the plain model (with publicly reconstructable output). If party A is now computing two separate first round messages, how does party B know which of them to use when computing its second round message? If B simply computes a second round message with respect to both, then since one may be mal-formed we are back to the original problem. One could try to have B *secret share* its input and compute a (first and) second round message with respect to each share. However, this immediately runs into issues if the functionality is computing on B's input in any way. But we observe that this outline, with additional ideas, can be made to work for a special type of functionality. Specifically, this motivates the relaxation from general-purpose 2PC to conditional disclosure of secrets (CDS) protocol.

Conditional Disclosure of Secrets (CDS). In CDS, there is no computation performed on sender's input m at all, and can thus be secret shared across two independent executions. However, the issue of preserving receiver privacy remains, since secret sharing the witness will be problematic. We circumvent the problem by simply requiring that the sender not have first round message at all! Therefore, an honest receiver does not have to respond to any potentially mal-formed sender message. In summary, then, we seek an instantiation of the following primitive.

- The receiver, on input a witness w, publishes a first round message $\mathsf{Com}(w)$.
- The parties decide to compute a CDS for circuit C.
- The sender, on input a message m, outputs a second round message $\mathsf{Enc}(m)$ that is computed with respect to $\mathsf{Com}(w)$.
- Simultaneously, the receiver outputs a second round message π_C, also computed with respect to $\mathsf{Com}(w)$.
- Given $\mathsf{Com}(w), \mathsf{Enc}(m)$, and π_C, anybody can recover m if $C(w) = 1$, and otherwise m is completely hidden.

Recently, Benhamouda and Lin [BL20] gave a construction (which they call "witness encryption for NIZK of commitment") that essentially satisfies the above syntax, except that it requires a CRS to be secure against malicious parties. While we seemingly have not made much progress, observe that we have significantly reduced the functionality, enough to make our initial idea work. In our scheme, the sender and the receiver will run two parallel copies of the above system, where CRSs are chosen by the receiver. Specifically, the receiver will send

$$(\mathsf{crs}_0, \mathsf{crs}_1, \mathsf{Com}_0(w), \mathsf{Com}_1(w))$$

together with a NIWI proof that at least one of the two copies is correctly computed. The sender will then respond with

$$(\mathsf{Enc}(m_0), \mathsf{Enc}(m_1)) \text{ such that } m_0 \oplus m_1 = m$$

and the receiver will simultaneously respond with both copies of the second round message $\pi_{C,0}$ and $\pi_{C,1}$. In terms of security, the NIWI guarantees that at least one of the two copies is correctly computed, which in turn implies that one of the shares of the message is hidden, if $C(w) \neq 1$.

Upgrading the Functionality. Now, the above gives a non-trivial two-party functionality that may be computed in two rounds in the plain model. We further observe that, due in part to the simplicity of the CDS functionality, the same techniques naturally extend to the *multi*-party setting. Here, we consider multiple receivers, each with a different input witness w_i and each associated with a different circuit C_i. A single sender can now additively secret share its message across all receiver commitments, so that m may only be recovered if $C_i(w_i) = 1$ for all i. In the next section, we show how this simple multi-party functionality can be used as a crucial building block for computing *all* multi-party functionalities in the plain model.

Before moving on we note that the initial construction given in [BL20] only supports computation of NC1 circuits, and they later upgrade their construction to support all polynomial-size circuits via the use of a randomized encoding with encoding in NC1 and a garbled circuit. Our construction uses similar techniques, starting with the same underlying building blocks as [BL20] and then tailoring this NC1 to P upgrade to our (multi-party, plain model) setting. Details may be found in Section 3.3.

Two Round Maliciously-Secure MPC To construct a two-round maliciously secure MPC protocol, we start with any generic two-round MPC protocol which is secure against *semi-malicious* adversaries. In short, semi-malicious adversaries are those who follow the protocol specification (like semi-honest adversaries) but may choose arbitrary randomness. Two-round MPC protocols such as [GS18,BL18a,AJJM20] provide security against such class of adversaries. However, an arbitrary malicious adversary might choose not to follow the protocol specification (e.g. by generating messages that are outside the support of honest distribution).

Challenge: Message Integrity. If we allow the adversary to behave arbitrarily, the aforementioned protocols no longer guarantee any meaningful notion of security. Well-studied techniques, such as requiring a zero-knowledge proof of "honest" behavior from all parties, does not work because such ZK proofs require at least 2 rounds [Pas03]. Therefore, transferring the second MPC message only after verifying the ZK proofs will end up requiring 3 rounds in the overall protocol. If, somehow, we could achieve some kind of "delayed-verification" then this problem would be solved. To realize this intuition, we will rely on our MCDS primitive. A natural approach would be to encrypt the second MPC messages of parties using MCDS so that they can be decrypted only if all parties behaved honestly in their first round. However, this intuition does not directly translate into a proof because of some key issues which we describe and address in the following.

From WI to Simulation Security. First, note that the MCDS only guarantees a witness-indistinguishability (WI) kind of security. In particular, it doesn't ensure that the witness (i.e. input and randomness) of parties remains hidden. All it ensures is that the choice, out of two possible witnesses (if they exist), remains hidden. Therefore, in order to leverage such WI-style security to provide a full-fledged ZK style guarantee, we will use the well-known FLS paradigm wherein we introduce a second "trapdoor" witness and require each party to prove (through MCDS) that either it behaved honestly in the first round OR it was successful in guessing the trapdoor. The trapdoor will be set up in a way so that a polynomially bounded adversary, in the real world, will not be able to guess the trapdoor and therefore will be forced to stick to the honest protocol. However, a super-polynomial time simulator, in the ideal world, would be able to guess the trapdoors and thereby generate the honest distribution *without* relying on the honest party witnesses (i.e. input and randomness).

To implement the aforementioned trapdoor-based solution, we rely on a special pair of commitment algorithms - com and Com. The idea is to have each party P_i generate a commitment $c_i = \mathsf{com}(0; r_i)$ using a uniformly random value r_i. Now the collection of all such n random values $\{r_i\}_{i \in [n]}$ will be used as a single trapdoor for all n parties. Concretely, each party P_i will be required to prove (through MCDS) that either there exists a valid witness w_i (encoding the semi-malicious MPC input and randomness) in its MCDS commitment (in Round 1) which is consistent with its first round semi-malicious MPC message OR that its MCDS commitment message contains the exact trapdoor values $\{r_i\}_{i \in [n]}$.

Malleability Attacks. Unfortunately, the above idea is not yet sufficient for achieving security due to the existence of different types of malleability attacks. For example, consider a scenario where the adversary $\mathcal{A}$, on receiving c_i^H from some (set of) honest party, "mauls" it into his own MCDS commitment value. If this happens, the second OR branch of the adversary's MCDS statement will be valid, and we won't be able to invoke the sender-security of the MCDS scheme to argue that the second round MPC message of honest parties is hidden. To handle this, we add a requirement that each party P_i must generate a commitment $C_i = \mathsf{Com}(0^{kn})$ in the first round and modify the second OR branch of the MCDS statement to additionally verify whether $C_i = \mathsf{Com}(\{r_i\}_{i \in [n]})$. The pair of commitment algorithms $(\mathsf{com}, \mathsf{Com})$ is designed so that any (implicit) information from c_i cannot be (efficiently) transferred to C_i. In other words, com is non-malleable w.r.t. to Com. In the real world, this will ensure that a polynomially-bounded adversary is unable to take the trapdoor branch of the MCDS statement. However, in the ideal world, the super-polynomial simulator will be able to do so by just guessing the trapdoor values.

A subtle issue that arises is the following: What happens if $\mathcal{A}$ just "copies" the exact same messages as that of the honest party? If this happens, he would be able to decrypt the second round MPC messages of honest parties just by using the exact same MCDS proof messages as that of honest parties. This is because the MCDS statement, along with the implicit witness in the copied first round MCDS message, of the adversary would be *exactly the same* as that of the honest party. Such attacks might be devastating because they might enable $\mathcal{A}$ to make his input "dependent" on the honest party's input. For example, consider a 2-party case where P_1 holds input x, P_2 holds input y, and they would like to securely compute $f(x,y)$. In such cases, a malleability attack might enable a corrupt P_2 to recover $f(x,x)$ with probability one. Note that such an attack is not allowed in the ideal world where each P_i sends its input to the functionality independently (of other parties). To thwart such attacks, we require Com to be a non-malleable commitment i.e. a commitment C_1 generated by honest party P_1 cannot be "mauled" into a related commitment C_2 by corrupt party P_2. From the protocol perspective, this ensures that an adversary which tries to copy the exact same messages as that of the honest party will be detected in the first round (as the non-malleable design of Com enforces each P_i to use a unique tag). From the perspective of security proof, this enables the simulator, in one of the hybrids, to switch from using the real inputs of honest parties to using the trapdoor witness in its C_i messages without letting the adversary also perform the same kind of switch.

Integrity of the Second Round. Although MCDS helps us conditionally transfer the second MPC message of honest parties, an adversary might still be able to "cheat" in his second round after behaving honestly in the first round. For example, an adversary generating "malformed" second round messages (i.e. messages outside the support of honest distribution) might be able to force honest parties into recovering an incorrect output without detection. Note that such attacks are not allowed by the real/ideal definition – in fact, in such a scenario, it is required that honest parties should be able to detect such an event and then abort. To fix this, we will use a type of (two-message) ZK argument, which we will again instantiate via a NIWI [GOS06a]. Essentially, each party will be required to prove, using NIWI, that either it is sending a well-formed second round message OR it has successfully guessed the trapdoor value $\{r_i\}_{i\in[n]}$ (which has already been set up in the first round as we described above).

Some Additional Challenges. Finally, we mention some of the details specific to the security proof of our protocol. Note that in a 2-round setting, rewinding is not an option for the simulator, and therefore the only way out is to correctly guess the adversary's actions in advance. This means that our simulator will make several (superpolynomially many) attempts to guess the adversary's trapdoor, and indistinguishability of hybrids will be conditioned on the event that the simulator was successful in correctly guessing *all* the trapdoors $\{r_i\}_{i\in[n]}$ (which includes the ones generated by the adversary). We note that it appears to be necessary to embed n trapdoors, one for each player, and allow the simulator (or any other player) to deviate from honest strategy if and only if it guessed the trapdoors of *all other players*. This, in turn, requires other primitives in the protocol to have a higher level of security than the total computation needed to guess all n trapdoors simultaneously. Concretely, assuming each c_i was created using γ bits of randomness in the com algorithm, then the simulator has a probability of $2^{-n\gamma}$ of being successful at the guess. Conditioned on this (very) low probability event, when we switch the value inside simulator's C_i^H from $0^{n\gamma}$ to the actual trapdoors $r_1||\ldots||r_n$, we would have to argue the independence of values inside adversary's C_i^M from the values inside C_i^H.[5] To enable this, we require that the non-malleable commitment scheme Com allows an advantage no better than $\mathsf{negl}(2^{n\gamma})$. We refer the reader to Section 2.4 for some plausible instantiations of such a primitive. Similarly, the other primitives in our protocol, such as MCDS and the semi-malicious MPC must also allow for an advantage no better than $\mathsf{negl}(2^{n\gamma})$.

At the same time, we would like to ensure that no adversary or set of colluding adversaries can copy the trapdoors $r_1||\ldots\ldots||r_n$, which include trapdoors used by honest parties. This means that we must ensure that commitments to r_i created according to the commitment scheme com cannot be mauled to generate commitments using the commitment scheme Com. Therefore, we interpret com and Com together as a "special" non-malleable commitment with $n+1$ tags, where commitments w.r.t. a special tag (say, the 0 tag) use at most γ bits of randomness and cannot be mauled to commitments via any other tag by a polynomial-sized circuit; and commitments with all non-zero tags are non-malleable w.r.t. each other with an advantage no better than $\mathsf{negl}(2^{n\gamma})$. We view identifying the right notion of non-malleability to instantiate our compiler as an important technical contribution of this work.

In Section 2.4, we provide instantiations for these special commitments in the setting of constant n (i.e. constant number of parties) based on sub-exponential time-lock puzzles and sub-exponential quantum hardness of the learning with errors (LWE) assumption. The restriction to constant n is due to the need for $\mathsf{negl}(2^{n\gamma})$ security, which is not satisfied by some existing constructions of non-interactive non-malleable commitments [LPS17,BL18b,KK19] for $n = \mathsf{poly}(\lambda)$. Nevertheless, we formulate an assumption on the hardness amplification of commitments (which is a variant of hardness amplifiability assumptions introduced in the context of non-malleable commitments by [BL18b]), and use this to instantiate special commitments for polynomial-sized tag spaces (and therefore, polynomially many parties) from sub-exponential falsifiable assumptions. We also provide a much simpler proof-of-concept instantiation from factoring-based adaptive one-way functions from [PPV08] and quantum hardness of the learning with errors (LWE) assumption. Due to the challenges outlined above, we believe that removing the need for special non-malleable commitments is likely to require new, possibly non-black-box, simulation techniques. However, we hope that future work will be able to simplify the assumptions on which special non-malleable commitments can be based by relying on other types of hardness.

[5] This is needed, for example, to ensure that the hybrid before switching to trapdoor is indistinguishable from the hybrid obtained after switching to trapdoor w.r.t an adversary who was unable to retrieve the Round 2 semi-malicious MPC message in the former hybrid (because of some dishonest behavior in the Round 1). We would like to avoid a scenario where such an adversary is actively trying to maul the honest party's C_i^H into its own C_i^M and therefore distinguishes the latter hybrid from the former one (by successfully retrieving the Round 2 MPC message in the latter but not the former).

Another interesting question is whether our protocols achieve a notion of angel-based security [PS04]. Angel based security allows the simulator as well as the adversary access to a super-polynomial resource called an "angel" which can perform a pre-defined task such as inverting a one-way function. Our simulation technique makes arguing angel-based security tricky: our simulator must guess the randomness that the adversary uses in his commitment c_i even before receiving these commitments from the adversary. Our simulator repeatedly runs the adversary until it guesses correctly, and it appears difficult to directly rely on an angel to make this guessing step easier. We believe that constructing two-round MPC satisfying angel-based or other forms of composable security is an interesting direction for future work.

2 Preliminaries

We say that a primitive satisfies (T, δ) security if the security definition holds for all $\mathsf{poly}(T)$ time adversaries with advantage at-most $\mathsf{negl}(\delta)$. Here T and δ can be arbitrary functions in the security parameter λ and all the honest parties should run in time $\mathsf{poly}(\lambda)$.

2.1 Non-Interactive Witness-Indistinguishable Proofs

We recall the notion of a non-interactive witness-indistinguishable (NIWI) proof system [GOS06b]. In [GOS06b] the authors showed how to construct such a NIWI based on standard hard problems over prime-order bilinear maps. A NIWI proof system is defined with respect to an NP language $\mathcal{L}$ with relation $\mathcal{R}$ and consists of the following efficient algorithms.

$\mathsf{NIWIProve}(x, w, \mathcal{R})$: On input a statement x, witness w, and relation $\mathcal{R}$, returns a proof π.

$\mathsf{NIWIVerify}(x, \pi, \mathcal{R})$: On input a statement x, proof π, and relation $\mathcal{R}$, the verification algorithm returns a bit $b \in \{0, 1\}$.

For correctness, we require that true statements always lead to accepting proofs.

Definition 1 (Correctness). *A NIWI proof system is correct if for all* $(w, x) \in \mathcal{R}$ *it holds that*

$$\mathsf{NIWIVerify}(x, \mathsf{NIWIProof}(x, w, \mathcal{R}), \mathcal{R}) = 1.$$

We require that the NIWI proof satisfies perfect soundness.

Definition 2 (Soundness). *A NIWI proof system is perfectly sound if for all* $x \notin \mathcal{L}$ *and for all proofs* π *it holds that*

$$\Pr\left[1 = \mathsf{NIWIVerify}(x, \pi, \mathcal{R})\right] = 0.$$

Finally, we require that the NIWI proof system satisfies the notion of computational witness-indistinguishability.

Definition 3 (Witness-Indistinguishability). *A NIWI proof system is witness indistinguishable if there exists a negligible function* negl *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) PPT adversaries* ADV*, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{ADV}(\pi) = b \\ \wedge (w_0, x) \in \mathcal{R} \wedge (w_1, x) \in \mathcal{R}\end{array}\middle| \begin{array}{l}(w_0, w_1, x) \leftarrow \mathsf{ADV}(1^\lambda) \\ b \leftarrow_{\$} \{0,1\} \\ \pi \leftarrow \mathsf{NIWIProve}(x, w_b, \mathcal{R})\end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

Groth et al. [GOS06b] showed that such a NIWI exists assuming the hardness of the DLin problem over bilinear maps.

Theorem 3 ([GOS06b]). *Let* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ *be a bilinear group where the DLin problem is hard. Then there exists a NIWI for NP.*

2.2 Garbled Circuit

We recall the definition of a garbling scheme for circuits [Yao86] (see Applebaum et al. [AIK04], Lindell and Pinkas [LP09] and Bellare et al. [BHR12] for a detailed proof and further discussion).

Definition 4 (Garbled Circuit). *A garbling scheme for circuits is a tuple of PPT algorithms* $(\mathsf{Garble}, \mathsf{GEval})$*.* Garble *is the circuit garbling procedure and* GEval *is the corresponding evaluation procedure. More formally:*

- $(\widetilde{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}\left(1^\lambda, C\right)$*:* Garble *takes as input a security parameter* 1^λ*, a circuit* C*, and outputs a* garbled circuit $\widetilde{C}$ *along with labels* $\{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}$*, where* n *is the length of the input to* C*.*
- $y \leftarrow \mathsf{GEval}\left(\widetilde{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}\right)$*: Given a garbled circuit* $\widetilde{C}$ *and a sequence of input labels* $\{\mathsf{lab}_{i,x_i}\}_{i\in[n]}$*,* GEval *outputs a string* y*.*

Correctness. For correctness, we require that for any circuit C *and input* $x \in \{0,1\}^n$ *we have that:*

$$\Pr\left[C(x) = \mathsf{GEval}\left(\widetilde{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}\right)\right] = 1$$

where $(\widetilde{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}\left(1^\lambda, C\right)$*.*

Security. For security, we require that there exists a PPT simulator GSim *such that for any circuit* C *and input* $x \in \{0,1\}^n$*, we have that*

$$\left(\widetilde{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}\right) \approx_c \mathsf{GSim}\left(1^\lambda, 1^{|C|}, 1^n, C(x)\right)$$

where $(\widetilde{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}\left(1^\lambda, C\right)$*.*

2.3 Randomized Encoding

We provide a definition of randomized encoding that is *perfectly correct*, *computationally private*, and has encoding in NC1. We follow the definition given in [BL20] which follows from [AIK05].

Definition 5 (Randomized Encoding). *Let $\mathcal{G}$ be a class of polynomial-size circuits. A computational randomized encoding scheme for $\mathcal{G}$ is a tuple of PPT algorithms* $(\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ *with the following syntax.*

- $\widehat{G} \coloneqq \mathsf{RE.Enc}(1^\lambda, G)$*: On input a security parameter and a circuit $G \in \mathcal{G}$, where $G : \{0,1\}^n \to \{0,1\}$, output a circuit $\widehat{G} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^p$. This procedure is deterministic.*
- $y \coloneqq \mathsf{RE.Dec}(1^\lambda, G, \widehat{y})$*: On input the security parameter, a circuit $C \in \mathcal{G}$, and the output $\widehat{y}$ of $\widehat{G}$, output the output y of G. This procedure is deterministic.*
- $\widehat{G} \leftarrow \mathsf{RE.Sim}(1^\lambda, G, y)$*: On input the security parameter, a circuit $G \in \mathcal{G}$, and an output $y \in \{0,1\}$, output a simulated randomized encoding $\widehat{G}$.*

Efficiency. We require that ℓ and p are polynomial in λ and in the size of G. We also require that $\widehat{G}$ is in NC1.

Perfect Correctness. For every $\lambda \in \mathbb{N}$, every circuit $G \in \mathcal{G}$, every input $x \in \{0,1\}^n$, and every string $r \in \{0,1\}^\ell$, we have that $\mathsf{RE.Dec}(1^\lambda, G, \widehat{G}(x,r)) = G(x)$, where $\widehat{G} \coloneqq \mathsf{RE.Enc}(1^\lambda, G)$.

Computational Privacy. For every circuit $G \in \mathcal{G}$ and every input $x \in \{0,1\}^n$, we have that

$$\left\{\widehat{G} \coloneqq \mathsf{RE.Enc}(1^\lambda, G), r \leftarrow \{0,1\}^\ell : \widehat{G}(v,r)\right\}_{\lambda \in \mathbb{N}} \approx_c \left\{\mathsf{RE.Sim}(1^\lambda, G, G(v))\right\}_{\lambda \in \mathbb{N}}.$$

2.4 Non-malleable Commitments

Non-malleability considers a man-in-the-middle MIM that receives a commitment to a message $m \in \{0,1\}^p$ and generates a new commitment $\widetilde{c}$. We say that MIM commits to $\perp$ if there does not exist any $(\widetilde{m}, \widetilde{r})$ such that $\widetilde{c} = \mathsf{com}(\widetilde{m}, \widetilde{r})$. Intuitively, the definition of non-malleability with respect to commitment requires that for any two messages $m_0, m_1 \in \{0,1\}^p$, the joint distributions of $(\mathsf{com}(m_0), \widetilde{m_0})$ and $(\mathsf{com}(m_1), \widetilde{m_1})$ are indistinguishable, where $\widetilde{m}_b$ is the message committed to by the MIM given $\mathsf{com}(m_b)$. We consider the case where the MIM gets a single committed message and generates a single commitment.

Definition 6 (One-to-One Non-malleable Commitments w.r.t. Commitment). *A non-interactive non-malleable (one-to-one) string commitment scheme with N tags consists of a probabilistic poly-time algorithm $\mathcal{C}$, that takes as input a message $m \in \{0,1\}^p$, randomness $r \in \{0,1\}^{\mathsf{poly}(\lambda)}$, and a $\mathsf{tag} \in [N]$, and outputs a commitment $\mathsf{com}_{\mathsf{tag}}(m; r)$. It is said to be non-malleable w.r.t. commitment if the following two properties hold:*

– **Binding.** *There do not exist* $m_0, m_1 \in \{0,1\}^p$, $r_0, r_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *and* $\mathsf{tag}_0, \mathsf{tag}_1 \in [N]$ *such that* $m_0 \neq m_1$ *and* $\mathsf{com}_{\mathsf{tag}_0}(m_0; r_0) = \mathsf{com}_{\mathsf{tag}_1}(m_1; r_1)$
– **One-to-One Non-malleability.** *For every pair of messages* $v_0, v_1 \in \{0,1\}^p$, *every pair of tags* $\mathsf{tag}, \widetilde{\mathsf{tag}}$, *every poly-size man-in-the-middle adversary* $\mathcal{A}$, *there exists a negligible function* $\mu(\cdot)$ *such that for all large enough* $\lambda \in \mathbb{N}$ *and all poly-size distinguishers* $\mathcal{D}$,

$$\Big| \Pr[\mathcal{D}(\mathcal{V}_0) = 1] - \Pr[\mathcal{D}(\mathcal{V}_1) = 1] \Big| = \mathsf{negl}(\lambda)$$

where for $\{b \in 0, 1\}$, *the distribution* $\mathcal{V}_b$ *is defined as follows:*
Sample $r \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\lambda)}$ *and set* $c = \mathsf{com}_{\mathsf{tag}}(m_b; r)$. *Let* $(\widetilde{c}, z) = \mathcal{A}(c)$. *If there exists* $\widetilde{\mathsf{tag}} \in [N] \setminus \mathsf{tag}, \widetilde{M} \in \{0,1\}^{p(\lambda)}$ *and* $\widetilde{r} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *such that* $\widetilde{c} = \mathsf{com}_{\widetilde{\mathsf{tag}}}(\widetilde{M}; \widetilde{r})$ *then* $\widetilde{m} = \widetilde{M}$, *otherwise set* $\widetilde{m} = \perp$. *The distribution* $\mathcal{V}_b$ *outputs* $(c, \widetilde{c}, \widetilde{m})$.

We will use a strengthened version of one-to-one non-malleable commitments, that we define next. Intuitively, we will require that there exist a special commitment (with say $\mathsf{tag} = 0^\kappa$), that uses only a very "short" string of randomness of size (say) λ. Looking ahead letting $n = \mathsf{poly}(\lambda)$ denote the number of parties in our MPC protocol, we will require commitments w.r.t. all non-zero tags to be $\mathsf{negl}(2^{n\gamma})$*-non-malleable w.r.t. each other (as opposed to* $\mathsf{negl}(\lambda)$*)*, for a γ that is described below. This property is formalized in Property 1 below. We will also need the special commitment (with say $\mathsf{tag} = 0^\kappa$) to satisfy the regular definition of (one-to-one) non-malleability w.r.t. all other tags, as formalized in Property 2 below.

Definition 7 **(n-Special One-to-One Non-malleable Commitments w.r.t. Commitment).** *A non-interactive non-malleable (one-to-one) string commitment scheme with* N *tags consists of a probabilistic poly-time algorithm* $\mathcal{C}$, *that takes as input a message* $m \in \{0,1\}^p$, *randomness* $r \in \{0,1\}^{\mathsf{poly}(\lambda)}$, *and a* $\mathsf{tag} \in [0, N]$, *and outputs a commitment* $\mathsf{com}_{\mathsf{tag}}(m; r)$. *It is said to be a special non-malleable commitment if the following three properties hold:*

– **Binding.** *There do not exist* $m_0, m_1 \in \{0,1\}^p$, $r_0, r_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *and* $\mathsf{tag}_0, \mathsf{tag}_1 \in [0, N]$ *such that* $m_0 \neq m_1$ *and* $\mathsf{com}_{\mathsf{tag}_0}(m_0; r_0) = \mathsf{com}_{\mathsf{tag}_1}(m_1; r_1)$
– **Property 1.** *For every pair of messages* $v_0, v_1 \in \{0,1\}^p$, *every pair of unequal tags* $\mathsf{tag} \in [1, N], \widetilde{\mathsf{tag}} \in [1, N]$, *every poly-size man-in-the-middle adversary* $\mathcal{A}$, *there exists a negligible function* $\mu(\cdot)$ *such that for all large enough* $\lambda \in \mathbb{N}$ *and all poly-size distinguishers* $\mathcal{D}$,

$$\Big| \Pr[\mathcal{D}(\mathcal{V}_0) = 1] - \Pr[\mathcal{D}(\mathcal{V}_1) = 1] \Big| = \mathsf{negl}(2^{\gamma \cdot n})$$

where γ *denotes the size of randomness used to commit to* λ*-bit messages with* $\mathsf{tag} = 0$, *and for* $\{b \in 0, 1\}$, *the distribution* $\mathcal{V}_b$ *is defined as follows:*
Sample $r \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\lambda)}$ *and set* $c = \mathsf{com}_{\mathsf{tag}}(m_b; r)$. *Let* $(\widetilde{c}, z) = \mathcal{A}(c)$. *If there exists* $\widetilde{\mathsf{tag}} \in [N] \setminus \mathsf{tag}, \widetilde{M} \in \{0,1\}^{p(\lambda)}$ *and* $\widetilde{r} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *such that*

$\widetilde{c} = \mathsf{com}_{\widetilde{\mathsf{tag}}}(\widetilde{M}; \widetilde{r})$ *then* $\widetilde{m} = \widetilde{M}$*, otherwise set* $\widetilde{m} = \perp$*. The distribution* $\mathcal{V}_b$ *outputs* $(c, \widetilde{c}, \widetilde{m})$*.*

– **Property 2.** *For every pair of messages* $v_0, v_1 \in \{0,1\}^p$*, every pair of tags* $\mathsf{tag}, \widetilde{\mathsf{tag}} \in [0, N]$ *such that* $\mathsf{tag} = 0$*, every poly-size man-in-the-middle adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\cdot)$ *such that for all large enough* $\lambda \in \mathbb{N}$ *and all poly-size distinguishers* $\mathcal{D}$*,*

$$\Big| \Pr[\mathcal{D}(\mathcal{V}_0) = 1] - \Pr[\mathcal{D}(\mathcal{V}_1) = 1] \Big| = \mathsf{negl}(\lambda)$$

where for $\{b \in 0, 1\}$*, the distribution* $\mathcal{V}_b$ *is defined as follows:*
Sample $r \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\lambda)}$ *and set* $c = \mathsf{com}_{\mathsf{tag}}(m_b; r)$*. Let* $(\widetilde{c}, z) = \mathcal{A}(c)$*. If there exists* $\widetilde{\mathsf{tag}} \in [N] \setminus \mathsf{tag}, \widetilde{M} \in \{0,1\}^{p(\lambda)}$ *and* $\widetilde{r} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *such that* $\widetilde{c} = \mathsf{com}_{\widetilde{\mathsf{tag}}}(\widetilde{M}; \widetilde{r})$ *then* $\widetilde{m} = \widetilde{M}$*, otherwise set* $\widetilde{m} = \perp$*. The distribution* $\mathcal{V}_b$ *outputs* $(c, \widetilde{c}, \widetilde{m})$*.*

We now describe different possible instantiations of such special non-malleable commitments. First, in the setting of constant tags, we obtain the following lemma by combining non-malleable commitments based on time-lock puzzles [LPS17], and quantum vs. classical hardness [KK19].

Lemma 1. *[LPS17,KK19] Assuming non-malleable commitments for constant-sized tag spaces based on the RSW time-lock puzzle family of assumptions [LPS17], and assuming sub-exponential quantum hardness of LWE, for every* constant c*, there exist* c*-special one-to-one non-malleable commitments w.r.t. commitment for tags in* $[0, n]$ *satisfying Definition 7.*

Next, for the setting of polynomially many parties/tags, we develop a pathway to building the desired special non-malleable commitments from falsifiable assumptions. To this end, we first generalize the notion of hardness amplifiability from [BL18b] to consider non-interactive commitments instead of one-way functions, and require an exponentially low guessing advantage.

Definition 8. *We will say that a family of perfectly binding bit commitments is* δ*-hardness amplifiable if for every polynomial-sized probabilistic adversary* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$*, every sufficiently large polynomial* ℓ *and sufficiently large* $\lambda \in \mathbb{N}$

$$\Pr_{\forall i \in [\ell], x_i \leftarrow \{0,1\}^\lambda, r_i \leftarrow \{0,1\}^*, c_i = \mathsf{com}(x_i; r_i)} [\mathcal{A}_\lambda(c_1, \ldots, c_\ell) = x_1 \oplus \ldots \oplus x_\ell)] \leq \frac{1}{2} + 2^{-\delta \ell(\lambda)}$$

We have the following lemma, that follows by carefully instantiating parameters and combining prior work.

Lemma 2. *[LPS17,BL18b,KK19] Assume that the following exist.*

– *Quantum polynomially-hard non-interactive commitments that satisfy Definition 8 with* $\delta > 0$*.*
– *Classically polynomially-hard non-interactive commitments that satisfy Definition 8 with* $\delta > 0$*, and can be inverted in quantum polynomial time.*

- *Sub-exponentially secure non-interactive commitment.*
- *Sub-exponentially secure one-message weak zero-knowledge [BL18b].*

Then for every polynomial $n = n(\lambda)$, n-special one-to-one non-malleable commitments w.r.t. commitment with tags in $[0, n]$ satisfying Definition 7 exist.

The proofs of both these lemmas, together with a simpler instantiation from adaptive one-way functions and QLWE, can be found in Appendix **??** in the full version.

In addition, we will rely on standard notions of MPC with superpolynomial simulation and MPC against semi-malicious adversaries. For completeness, formal definitions can be found in Appendix **??** in the full version.

3 Multi-Party Conditional Disclosure of Secrets

In the following we define and construct a multi-party conditional disclosure of secrets protocol in two rounds, from standard assumptions over bilinear maps. Our protocol is (i) in the plain model and (ii) delayed-statement. Our construction is for general polynomial-size circuits, and satisfies computational sender and computational receiver security. We additionally provide a construction for NC1 circuits that satisfies *perfect* sender security in **??** in the full version.

3.1 Definition

A (delayed statement) multi-party conditional disclosure of secrets (MCDS) protocol is a 2-round protocol consisting of a single sender S and a set $\mathbb{R}$ of n receivers - $\{\mathsf{R}_1, \ldots, \mathsf{R}_n\}$. The sender holds a private message m whereas each receiver holds a private witness w_i. Additionally, the sender shares a (delayed) statement x_i with each R_i before the second round begins. If each of the n witnesses are valid witnesses to the corresponding statements x_i, then all the n receivers obtain m. However, if there exists $x_i \notin \mathcal{L}$, then m remains hidden from all the receivers.

More formally, an MCDS protocol is defined with respect to an NP language $\mathcal{L}$ with relation $\mathcal{R}$ and consists of the following algorithms.

$\mathsf{Com}(1^\lambda, w_i, i)$: On input the security parameter 1^λ and a witness w_i, the commitment algorithm returns the commitment c_i and a trapdoor t_i.

$\mathsf{E}((c_1, \ldots, c_n), (x_1, \ldots, x_n), m)$: On input n commitments $(c_1, \ldots, c_n)$, n statements $(x_1, \ldots, x_n)$, and a message m, the encryption algorithm returns a ciphertext d.

$\mathsf{Prove}(t_i, x_i)$: On input a trapdoor t_i and a statement x_i, the proving algorithm returns a decryption share p_i.

$\mathsf{Rec}(d, (p_1, \ldots, p_n))$: On input a ciphertext d and n decryption shares $(p_1, \ldots, p_n)$, the reconstruction algorithm returns a message m.

For correctness, we require that the message is always transmitted if all of the receivers commit to the correct witness.

Definition 9 (Correctness). *An MCDS protocol is correct if for all* $\lambda \in \mathbb{N}$*, all* $n \in \mathsf{poly}(\lambda)$*, all* $(w_i, x_i) \in \mathcal{R}$*, all* $m \in \{0,1\}$*, and all* (c_i, t_i) *in the support of* $\mathsf{Com}(1^\lambda, w_i)$*, it holds that*

$$\mathsf{Rec}(\mathsf{E}((c_1,\ldots,c_n),(x_1,\ldots,x_n),m),\mathsf{Prove}(t_1,x_1),\ldots,\mathsf{Prove}(t_n,x_n)) = m.$$

Sender security requires that the message is computationally hidden if at least one of the statements is false.

Definition 10 (Sender Security). *An MCDS protocol satisfies sender security if there exists a negligible function* negl *such that for all* $\lambda \in \mathbb{N}$*, all* $n \in \mathsf{poly}(\lambda)$*, and all (stateful) PPT adversaries* ADV*, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{ADV}(d) = b \\ \wedge\ \exists i : x_i \notin \mathcal{L} \end{array} \middle| \begin{array}{l} (m_0, m_1, c_1, \ldots, c_n, x_1, \ldots, x_n) \leftarrow \mathsf{ADV}(1^\lambda) \\ b \leftarrow_{\$} \{0,1\} \\ d \leftarrow \mathsf{E}((c_1,\ldots,c_n),(x_1,\ldots,x_n),m_b) \end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

Receiver security is analogous to witness indistinguishability and says that any adversary cannot distinguish between the commitment of two valid witnesses, even after seeing a proof for a statement of his choice. The following property in particular implies security for any receiver, even if the adversary corrupts every other party in the system.

Definition 11 (Receiver Security). *An MCDS protocol satisfies receiver security if there exists a negligible function* negl *such that for all* $\lambda \in \mathbb{N}$*, all* $n \in \mathsf{poly}(\lambda)$*, and all (stateful) PPT adversaries* ADV*, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{ADV}(\pi) = b \\ \wedge\ (w_0, x) \in \mathcal{R} \wedge (w_1, x) \in \mathcal{R} \end{array} \middle| \begin{array}{l} (w_0, w_1) \leftarrow \mathsf{ADV}(1^\lambda) \\ b \leftarrow_{\$} \{0,1\} \\ (c, t) \leftarrow \mathsf{Com}(1^\lambda, w_b) \\ x \leftarrow \mathsf{ADV}(c) \\ \pi \leftarrow \mathsf{Prove}(t, x) \end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

We say that the MCDS satisfies *reusable* receiver security if the adversary is additionally given access to a proving oracle $\mathsf{Prove}(t, \cdot)$ that can be queried on any statement x such that $(w_0, x) \in \mathcal{R}$ and $(w_1, x) \in \mathcal{R}$.

General Access Structures. It is worth mentioning that we define and consider only the AND access structure across all statements, i.e. the message is revealed if (and only if) *all* statements are true. This simple access structure will be sufficient for our purposes, however one could image scenarios where more complex access structures are needed. Although we do not elaborate on it, both our definitions and our constructions naturally extend to the more general settings.

3.2 Witness Encryption for Dual Mode Commitments

We recall the notion of dual-mode commitment from [BL20]. We first define the basic interfaces.

$\mathsf{DualSetupB}(1^\lambda)$: On input the security parameter, the setup algorithm (in binding mode) returns a common reference string crs.

$\mathsf{DualSetupH}(1^\lambda)$: On input the security parameter, the setup algorithm (in hiding mode) returns a common reference string crs and a trapdoor τ.

$\mathsf{DualCom}(\mathsf{crs}, m; r)$: On input the common reference string crs, a message m, and some random coins r, the commitment algorithm returns a commitment com.

$\mathsf{DualProof}(\mathsf{crs}, \mathsf{com}, r, C, y)$: On input a common reference string crs, a commitment com, random coins r, circuit C, and output y, the proof algorithm returns a proof π.

$\mathsf{DualVerify}(\mathsf{crs}, \mathsf{com}, \pi, C, y)$: On input a common reference string crs, a commitment com, a proof π, a circuit C, and an output y, the verification algorithm returns a bit $b \in \{0, 1\}$.

The scheme satisfies perfect correctness in the following sense.

Definition 12 (Correctness). *A dual-mode commitment scheme is perfectly correct if for all $\lambda \in \mathbb{N}$, all crs in the support of $\mathsf{DualSetupB}$ (or $\mathsf{DualSetupH}$), all messages m, all random coins r, all circuits C, it holds that*

$$1 = \mathsf{DualVerify}(\mathsf{crs}, \mathsf{com}, \mathsf{DualProof}(\mathsf{crs}, \mathsf{com}, r, C, C(m)), C, C(m)).$$

where $\mathsf{com} = \mathsf{DualCom}(\mathsf{crs}, m; r)$.

We require that the scheme satisfies setup indistinguishability, i.e. it is hard to distinguish between common reference strings sampled in binding or hiding mode.

Definition 13 (Setup Indistinguishability). *A dual-mode commitment scheme satisfies setup indistinguishability if there exists a negligible function* negl *such that for all $\lambda \in \mathbb{N}$ and all (stateful) PPT adversaries* ADV, *it holds that*

$$\Pr\left[\mathsf{ADV}(\mathsf{crs}) = b \,\middle|\, \begin{array}{l} b \leftarrow_\$ \{0,1\} \\ \mathsf{crs} \leftarrow \mathsf{DualSetupB}(1^\lambda) \text{ if } b = 0 \\ (\mathsf{crs}, \tau) \leftarrow \mathsf{DualSetupH}(1^\lambda) \text{ if } b = 1 \end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

We require the strong notion of perfect soundness when the common reference string is sampled in binding mode.

Definition 14 (Soundness). *A dual-mode commitment scheme satisfies perfect soundness if for all $\lambda \in \mathbb{N}$, all crs in the support of $\mathsf{DualSetupB}(1^\lambda)$, all messages m, all random coins r, all com in the support of $\mathsf{DualCom}(\mathsf{crs}, m; r)$, all circuits C, all $y \neq C(m)$, and all proofs π it holds that*

$$\Pr[1 = \mathsf{DualVerify}(\mathsf{crs}, \mathsf{com}, \pi, C, y)] = 0.$$

We further require that, if the common reference string is sampled in hiding mode, then proofs can be perfectly simulated.

Definition 15 (Zero-Knowledge). *A dual-mode commitment satisfies zero-knowledge if there exists a negligible function* negl *and a PPT simulator* $(\mathsf{Sim}_{\mathsf{com}}, \mathsf{Sim}_\pi)$ *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) PPT adversaries* ADV*, it holds that*

$$\Pr\left[\mathsf{ADV}(\mathsf{com})^{\mathsf{Prove}(\cdot)} = b \;\middle|\; \begin{array}{l} (\mathsf{crs}, \tau) \leftarrow \mathsf{DualSetupH}(1^\lambda) \\ m \leftarrow \mathsf{ADV}(\mathsf{crs}, \tau) \\ b \leftarrow_{\$} \{0,1\} \\ \mathsf{com} \leftarrow \mathsf{DualCom}(\mathsf{crs}, m; r) \text{ if } b = 0 \\ (\mathsf{com}, \alpha) \leftarrow \mathsf{Sim}_{\mathsf{com}}(\mathsf{crs}, \tau) \text{ if } b = 1 \end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda)$$

where $\mathsf{Prove}(C) = \mathsf{DualProof}(\mathsf{crs}, m, r, C, C(m))$ *if* $b = 0$ *and* $\mathsf{Prove}(C) = \mathsf{Sim}_\pi(\tau, \alpha, C, C(m))$ *if* $b = 1$.

Bit Commitments. We remark that, unless differently specified, in this work we always consider commitments to single bits. The construction of [BL20] is a bit commitment, although not explicitly defined this way. Specifically we are going to use the property that the hiding of any commitment to n bits can be broken in time $2^\lambda \cdot n$, where λ is the security parameter of the commitment scheme.

Witness Encryption. We augment the syntax of the dual-mode commitment with a witness encryption algorithm. This allows anyone to encrypt a message with respect to a circuit C, which can be decrypted publicly with a proof π that certifies that the commitment message m satisfies $C(m) = y$. The formal syntax is given below.

$\mathsf{WEnc}(\mathsf{crs}, \mathsf{com}, C, y, m')$: On input a common reference string crs, a commitment com, a circuit C, an output y, and a message m', the encryption algorithm returns a ciphertext c.

$\mathsf{WDec}(\mathsf{crs}, \mathsf{com}, \pi, c, y)$: On input a common reference string crs, a commitment com, a proof π, a ciphertext c, and an output y, the decryption algorithm returns a message m'.

We define correctness below.

Definition 16 (Correctness). *A witness encryption for a dual-mode commitment is correct if for all* $\lambda \in \mathbb{N}$*, all* crs *in the support of* $\mathsf{DualSetupB}$ *(or* $\mathsf{DualSetupH}$*), all messages* m, m'*, all random coins* r*, and all circuits* C *it holds that*

$$\mathsf{WDec}(\mathsf{crs}, \mathsf{com}, \mathsf{DualProof}(\mathsf{crs}, \mathsf{com}, r, C, C(m)), \mathsf{WEnc}(\mathsf{crs}, \mathsf{com}, C, C(m), m')) = m',$$

where $\mathsf{com} = \mathsf{DualCom}(\mathsf{crs}, m; r)$.

Furthermore, we define semantic security. We require a strong notion where the message is perfectly hidden even to the eyes of an unbounded adversary.

Definition 17 (Semantic Security). *A witness encryption for a dual-mode commitment is semantically secure if for all (stateful) unbounded adversaries*

ADV *it holds that*

$$\Pr\left[\mathsf{ADV}(c)=b \;\middle|\; \begin{array}{l} \rho \leftarrow \mathsf{ADV}(1^\lambda) \\ \mathsf{crs} \leftarrow \mathsf{DualSetupB}(1^\lambda;\rho) \\ (m,r,C,y,m'_0,m'_1) \leftarrow \mathsf{ADV}(\mathsf{crs}) \\ \mathsf{com} \leftarrow \mathsf{DualCom}(\mathsf{crs},m;r) \\ b \leftarrow_{\$} \{0,1\} \\ c \leftarrow \mathsf{WEnc}(\mathsf{crs},\mathsf{com},C,C(m),m_b) \text{ if } C(m) \neq y \\ c \leftarrow \perp \text{ otherwise} \end{array}\right] = 1/2.$$

We recall the main theorem statement from [BL20], which says that a dual-mode commitment with witness encryption for NC1 circuit exists assuming the hardness of the SXDH problem over bilinear maps.

Theorem 4 ([BL20]). *Let* $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T)$ *be a bilinear group where the SXDH problem is hard. Then there exists a dual-mode commitment scheme with witness encryption for NC1 circuits.*

3.3 Construction of MCDS

In the following we describe our construction of MCDS for polynomial-size circuits. As the underlying building blocks we assume the dual-mode commitment with witness encryption from [BL20], NIWI proofs from [GOS06b], computational randomized encodings (Definition 5), and garbled circuits (Definition 4).

Let $\mathcal{U}=\{U_\lambda:\{0,1\}^{h(\lambda)}\times\{0,1\}^{k(\lambda)}\to\{0,1\}\}_{\lambda\in\mathbb{N}}$ be the family of verification circuits for an NP language $\mathcal{L}$, where each U_λ takes as input an instance $x\in\{0,1\}^{h(\lambda)}$ and a witness $w\in\{0,1\}^{k(\lambda)}$, and outputs a bit indicating acceptance or rejection. For any fixed instance x, we consider the circuit $U_\lambda[x]:\{0,1\}^{k(\lambda)}\to\{0,1\}$ that just takes as input a witness w. Let $\ell(\lambda)$ and $p(\lambda)$ be parameters for computing a randomized encoding of $U_\lambda[x]$. That is, $\mathsf{RE.Enc}(1^\lambda,U_\lambda[x])$ outputs $\widehat{U}_\lambda[x]=\left(\widehat{U}_\lambda[x]_1,\ldots,\widehat{U}_\lambda[x]_{p(\lambda)}\right)$, where each $\widehat{U}_\lambda[x]_i:\{0,1\}^{k(\lambda)}\times\{0,1\}^{\ell(\lambda)}\to\{0,1\}$. In the construction below, define $\ell:=\ell(\lambda), p:=p(\lambda)$, and $U:=U_\lambda$. Let n be the number of receivers.

- $\mathsf{Com}(1^\lambda,w_i)$:
  - Sample two common reference strings

    $$\mathsf{crs}_{i,0}\leftarrow\mathsf{DualSetupB}(1^\lambda),\mathsf{crs}_{i,1}\leftarrow\mathsf{DualSetupB}(1^\lambda)$$

    in binding mode for the dual-mode commitment.
  - Compute two commitments

    $$\mathsf{com}_{i,0}=\mathsf{DualCom}(\mathsf{crs}_{i,0},(w_i,r_{i,0});s_{i,0}),\mathsf{com}_{i,1}=\mathsf{DualCom}(\mathsf{crs}_{i,1},(w_i,r_{i,1});s_{i,1}),$$

    where $r_{i,0},r_{i,1}\leftarrow\{0,1\}^p$ and $s_{i,0},s_{i,1}\leftarrow\{0,1\}^\lambda$.
  - Compute the NIWI proof

    $$\widetilde{\pi}_i\leftarrow\mathsf{NIWIProve}\left((z_i,0,w_i,s_{i,0}),\left\{\exists(z_i,b_i,w_i,s_i):\begin{array}{l}\mathsf{crs}_{i,b}=\mathsf{DualSetupB}(1^\lambda;z_i)\wedge\\ \mathsf{com}_{i,b}=\mathsf{DualCom}(\mathsf{crs}_{i,b},w_i;s_i)\end{array}\right\}\right).$$

- Return $c_i = (\mathsf{crs}_{i,0}, \mathsf{crs}_{i,1}, \mathsf{com}_{i,0}, \mathsf{com}_{i,1}, \widetilde{\pi}_i)$ and $t_i = (w_i, r_{i,0}, r_{i,1}, s_{i,0}, s_{i,1})$.

– $\mathsf{E}((c_1, \ldots, c_n), (x_1, \ldots, x_n), m)$:
  - Verify all of the NIWI proofs contained in the commitments, i.e. check whether for all $i = 1 \ldots n$ it holds that

$$1 = \mathsf{NIWIVerify}\left(\widetilde{\pi}_i, \left\{\exists (z_i, b_i, w_i, s_i) : \begin{array}{l} \mathsf{crs}_{i,b} = \mathsf{DualSetupB}(1^\lambda; z_i) \land \\ \mathsf{com}_{i,b} = \mathsf{DualCom}(\mathsf{crs}_{i,b}, w_i; s_i) \end{array}\right\}\right)$$

    and abort if this is not the case.
  - Compute a $2n$-out-of-$2n$ secret sharing $\{m_{i,a}\}_{i \in [n], a \in \{0,1\}}$ of m.
  - Define the circuit $f[i,a] : \{0,1\}^p \to \{m_{i,a}, \perp\}$ to take as input $\widehat{y}_i$ and output $m_{i,a}$ if $\mathsf{RE.Dec}(1^\lambda, U[x_i], \widehat{y}_i) = 1$, and otherwise output $\perp$.
  - For each $i \in [n], a \in \{0,1\}$, compute $(\widetilde{f}[i,a], \{\mathsf{lab}[i,a]_{j,b}\}_{j \in [p], b \in \{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, f[i,a])$.
  - For each $i \in [n]$, let $\left(\widehat{U}[x_i]_1, \ldots, \widehat{U}[x_i]_p\right) \coloneqq \mathsf{RE.Enc}(1^\lambda, U[x_i])$.
  - For each $i \in [n], a \in \{0,1\}, j \in [p]$, compute

$$d_{i,a,j,0} = \mathsf{WEnc}(\mathsf{crs}_{i,a}, \mathsf{com}_{i,a}, \widehat{U}[x_i]_j, \mathsf{lab}[i,a]_{j,0}, 0),$$
$$d_{i,a,j,1} = \mathsf{WEnc}(\mathsf{crs}_{i,a}, \mathsf{com}_{i,a}, \widehat{U}[x_i]_j, \mathsf{lab}[i,a]_{j,1}, 1)$$

  - Output

$$d = \left(\left\{\widetilde{f}[i,a], \{d_{i,a,j,b}\}_{j,b}\right\}_{i,a}\right).$$

– $\mathsf{Prove}(t_i, x_i)$:
  - Parse t_i as $(w_i, r_{i,0}, r_{i,1}, s_{i,0}, s_{i,1})$.
  - Compute $\widehat{y}_{i,0} \coloneqq \widehat{U}[x_i](w_i, r_{i,0})$ and for each $j \in [p]$, compute

$$\pi_{i,j,0} \leftarrow \mathsf{DualProof}(\mathsf{crs}_{i,0}, \mathsf{com}_{i,0}, s_{i,0}, \widehat{U}[x_i]_j, (\widehat{y}_{i,0})_j).$$

  - Compute $\widehat{y}_{i,1} \coloneqq \widehat{U}[x_i](w_i, r_{i,1})$ and for each $j \in [p]$, compute

$$\pi_{i,j,1} \leftarrow \mathsf{DualProof}(\mathsf{crs}_{i,1}, \mathsf{com}_{i,1}, s_{i,1}, \widehat{U}[x_i]_j, (\widehat{y}_{i,1})_j).$$

  - Output $\left(\widehat{y}_{i,0}, \widehat{y}_{i,1}, \{\pi_{i,j,0}\}_{j \in [p]}, \{\pi_{i,j,1}\}_{j \in [p]}\right)$.

– $\mathsf{Rec}(d, (p_1, \ldots, p_n))$:
  - Parse d as $\left(\left\{\widetilde{f}[i,a], \{d_{i,a,j,b}\}_{j,b}\right\}_{i,a}\right)$ and each p_i as $\left(\widehat{y}_{i,0}, \widehat{y}_{i,1}, \{\pi_{i,j,0}\}_{j \in [p]}, \{\pi_{i,j,1}\}_{j \in [p]}\right)$.
  - For each $i \in [n], a \in \{0,1\}, j \in [p]$, compute

$$\mathsf{lab}[i,a]_j \leftarrow \mathsf{WDec}(\mathsf{crs}_{i,a}, \mathsf{com}_{i,a}, \pi_{i,j,a}, d_{i,a,j,0}, (\widehat{y}_{i,a})_j).$$

  - For each $i \in [n], a \in \{0,1\}$, compute $m_{i,a} = \mathsf{GEval}(\widetilde{f}[i,a], \{\mathsf{lab}[i,a]_j\}_j)$.
  - Ouptut $m = \bigoplus_{i,a} m_{i,a}$.

Sender Security. We show that our MCDS protocol satisfies computational sender security.

Theorem 5 (Sender Security). *Assuming a dual-mode commitment with witness encryption (Section 3.2), NIWI proofs (Section 2.1), computational randomized encodings (Definition 5), and garbled circuits (Definition 4), the MCDS protocol* $(\mathsf{Com}, \mathsf{E}, \mathsf{Prove}, \mathsf{Rec})$ *as described above satisfies computational sender security. These primitives follow from the existence of a bilinear group where the SXDH problem is hard and the existence of a bilinear group where where the DLIN problem is hard.*

We will actually prove the following lemma, which immediatedly implies the theorem due to the perfect soundness of NIWI. The particular property defined by the lemma will be useful later in our MPC construction.

Lemma 3. *For all (stateful) unbounded adversaries* ADV*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}\mathsf{ADV}(d) = b \\ \wedge\ \exists (i,a) : \mathsf{crs}_{i,a} \in \mathsf{DualSetupB}(1^\lambda) \\ \wedge\ \mathsf{com}_{i,a} \in \mathsf{DualCom}(\mathsf{crs}_{i,a}, (w_i, r_i)) \\ \wedge\ (w_i, x_i) \notin \mathcal{R}\end{array}\ \middle|\ \begin{array}{l}(m_0, m_1, c_1, \ldots, c_n, x_1, \ldots, x_n) \leftarrow \mathsf{ADV}(1^\lambda) \\ b \leftarrow_{\$} \{0,1\} \\ d \leftarrow \mathsf{E}((c_1, \ldots, c_n), (x_1, \ldots, x_n), m_b)\end{array}\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

Proof. We will show that an adversary ADV contradicting the lemma can be used to break security of the garbled circuit.

Fix the message $(m_0, m_1, c_1, \ldots, c_n, x_1, \ldots, x_n)$ output by ADV for which it has the best advantage, and let (i, a) be the associated tuple guaranteed by the lemma statement. Recall that the encryption of $m \in \{m_0, m_1\}$ that ADV sees consists of $2n$ garbled circuits along with witness encryptions of each of the labels. Let $\mathcal{D}_b$ be the distribution that samples an encryption of m_b. It suffices to show that for each $b \in \{0,1\}$, $\mathcal{D}_b$ is indistinguishable from a distribution $\mathcal{E}_b$ that is identical to $\mathcal{D}_b$ except that the circuit $f[i,a]$ that is garbled has 0 hard-coded rather than the share $m_{i,a}$. This follows because $\mathcal{E}_0$ is identically distributed to $\mathcal{E}_1$, since the collection of shares other than $m_{i,a}$ are uniformly random, regardless of the message.

Now, by the perfect soundness of the witness encryption (for NC1), we know that for each $j \in [p]$, at least one of

$$d_{i,a,j,0} = \mathsf{WEnc}(\mathsf{crs}_{i,a}, \mathsf{com}_{i,a}, \widehat{U}[x_i]_j, \mathsf{lab}[i,a]_{j,0}, 0),$$
$$d_{i,a,j,1} = \mathsf{WEnc}(\mathsf{crs}_{i,a}, \mathsf{com}_{i,a}, \widehat{U}[x_i]_j, \mathsf{lab}[i,a]_{j,1}, 1)$$

is a perfectly hiding encryption. In particular, the only labels that ADV will be able to decrypt are those that correspond to the input $\widehat{y}_i = \widehat{U}[x_i](w_i, x_i)$. Since $(w_i, x_i) \notin \mathcal{R}$, by the perfect correctness of the randomized encoding, we know that $\mathsf{RE.Dec}(1^\lambda, U[x_i], \widehat{y}_i) = 0$, and thus that $f[i,a](\widehat{y}_i) = \bot$, regardless of which value $m_{i,a}$ is hard-coded. Thus, for each $b \in \{0,1\}$ there exists a reduction $\mathcal{R}_b$

that takes as input either i) a garbling of $f[i, a]$ with $m_{i,a}$ hard-coded along with labels corresponding to $\widehat{y}$, and perfectly simulates $\mathcal{D}_b$, or ii) a garbling of $f[i, a]$ with 0 hard-coded along with labels corresponding to $\widehat{y}$, and perfectly simulates $\mathcal{E}_b$. But by the security of the garbled circuit, the distributions seen by $\mathcal{R}_b$ are computationally indistinguishable, since they can both be simulated by $\mathsf{GSim}(1^\lambda, 1^{|f|}, 1^{p \cdot n}, \perp)$.

Receiver Security. We show that our MCDS protocol satisfies computational receiver security.

Theorem 6 (Receiver Security). *Assuming a dual-mode commitment with witness encryption (Section 3.2), NIWI proofs (Section 2.1), computational randomized encodings (Definition 5), and garbled circuits (Definition 4), the MCDS protocol* $(\mathsf{Com}, \mathsf{E}, \mathsf{Prove}, \mathsf{Rec})$ *as described above satisfies computational receiver security. These primitives follow from the existence of a bilinear group where the SXDH problem is hard and the existence of a bilinear group where where the DLIN problem is hard.*

Proof. We prove the theorem by defining a series of hybrids, then we argue that each pair of hybrids is indistinguishable by any PPT adversary.

- Hyb_0: This is the original experiment, with the bit of the challenger set to 0, i.e. the commitment c is always computed as $\mathsf{Com}(1^\lambda, w_0)$.
- Hyb_1: This hybrid is identical to the previous one, except that in the computation of the algorithm Com, the common reference string $\mathsf{crs}_{i,1}$ is computed in hiding mode, i.e. $(\mathsf{crs}_{i,1}, \tau_1) \leftarrow \mathsf{DualSetupH}(1^\lambda)$. Computational indistinguishability follows from the setup indistinguishability of the dual-mode commitment.
- Hyb_2: In this hybrid we further modify the Com algorithm to compute a simulated commitment $(\mathsf{com}_{i,1}, \alpha_1) \leftarrow \mathsf{Sim}_{\mathsf{com}}(\mathsf{crs}_{i,1}, \tau_1)$ and we switch to simulated proofs $\pi_{i,j,1} \leftarrow \mathsf{Sim}_\pi(\tau_1, \alpha_1, \widehat{U}[x_i]_j, (\widehat{y}_{i,1})_j)$. By the zero-knowledge property of the dual mode commitment, this modification is computationally indistinguishable to the eyes of the adversary.
- Hyb_3: In this hybrid we switch $\widehat{y}_{i,1}$ to be computed as $\mathsf{RE.Sim}(1^\lambda, U[x_i], U[x_i](w_0))$. This is indistinguishable due to the computational privacy of the randomized encoding.
- Hyb_4: In this hybrid we switch $\widehat{y}_{i,1}$ to be computed as $\mathsf{RE.Sim}(1^\lambda, U[x_i], U[x_i](w_1))$. This is perfectly indistinguishable by the definition of receiver security, which requires that $U[x_i](w_0) = U[x_i](w_1)$.
- Hyb_5: In this hybrid, we no longer simulate the commitment, computing $\mathsf{com}_{i,1} \leftarrow \mathsf{DualCom}(\mathsf{crs}_{i,1}, w_1; s_{i,1})$ and then computing the proofs $\pi_{i,j,1}$ honestly. Thus this modification is computationally indistinguishable by another invocation of the zero-knowledge property of the dual-mode commitment.
- Hyb_6: Here we compute $\mathsf{crs}_{i,1}$ back in binding mode, i.e. $\mathsf{crs}_{i,1} \leftarrow \mathsf{DualSetupB}(1^\lambda)$. Indistinguishability follows from the setup indistinguishability of the dual-mode commitment.

- Hyb_7: In this hybrid we switch the branch of the NIWI proof, i.e. we compute the NIWI proof using the witness $(z_{i,1}, 1, w_1, s_{i,1})$, instead of $(z_{i,0}, 0, w_0, s_{i,0})$. The rest of the algorithms are unchanged. Note that both witnesses are valid for the given statement and therefore indistinguishability follows from the witness-indistinguishability of the NIWI proof.
- $\mathsf{Hyb}_8 \dots \mathsf{Hyb}_{13}$: These hybrids are defined identically to $\mathsf{Hyb}_1 \dots \mathsf{Hyb}_6$ except that we simulate $\mathsf{crs}_{i,0}$ and we switch the witness used in $\mathsf{com}_{i,0}$ to be w_1, then we revert the change in the sampling of the common reference string. The arguments to show indistinguishability of each pair of hybrids are identical.
- Hyb_{14}: In this hybrid we switch again th branch of the NIWI proof, i.e. we compute the proof using the witness $(z_{i,0}, 0, w_1, s_{i,0})$ instead of $(z_{i,1}, 1, w_1, s_{i,1})$. Indistinguishability follows from an invocation of the computational witness-indistinguishability of the NIWI proof.

Observe that the distribution induced by Hyb_{14} is identical to that of Hyb_1 except that the committed message is fixed to w_1, instead of w_0. By the above analysis, $\mathsf{Hyb}_1 \approx_c \mathsf{Hyb}_0$ are computationally indistinguishable, which concludes our proof.

We note that MCDS with sub-exponential security follows by instantiating the underlying hardness assumptions (SXDH and DLin over bilinear maps) with their sub-exponentially secure versions. This is because all our security reductions in the MCDS construction can be observed to run in time $p(\lambda, T')$ for a fixed polynomial $p(\cdot)$, where λ is the security parameter and T' is the running time of the MCDS adversary. This will lead to a contradiction against T-security of the underlying hardness assumption for any subexponential T. We will require MCDS with sub-exponential security in our construction of the two round maliciously secure MPC.

Theorem 7 (Sub-exponential Sender Security). *Assuming sub-exponentially secure garbled circuits (i.e. one-way functions), the MCDS protocol* $(\mathsf{Com}, \mathsf{E}, \mathsf{Prove}, \mathsf{Rec})$ *as described above satisfies sub-exponential sender security.*

Theorem 8 (Sub-exponential Receiver Security). *Assuming sub-exponential SXDH and DLin, the MCDS protocol* $(\mathsf{Com}, \mathsf{E}, \mathsf{Prove}, \mathsf{Rec})$ *as described above satisfies sub-exponential receiver security.*

Reusable Receiver Security. Although we do not explicitly construct it, we note that the above scheme can be easily lifted to the reusable settings, i.e. where the committed can be reused for polynomially-many instances of the second round (possibly with different messages and for different statments). The only subtlety that we need to address is that the randomness used to compute the randomized encoding cannot be hardwired in the commitment, instead it must be sampled using a PRF where the key is included in the commitment and the input is public. The only constraint that we impose on the PRF is that it must be computable by an NC1 circuit, which can be instantiated from a variety of assumptions (e.g. DDH [NR97] or LWE [BP14]).

4 Two Round Malicious MPC

We assume the existence of:

- A non-interactive witness-indistinguishable proof satisfying Definition 3.
- A special non-interactive non-malleable commitment NMCom satisfying Definition 7.
- A two-round semi-malicious MPC protocol satisfying Definition **??**.
- A multi-party CDS mCDS discussed in Section 3, satisfying Definitions 10 and 11.

We will use $\mathsf{mCDS}^{(i)}$, to indicate an mCDS session where P_i is the sender and all other parties $\{P_j\}_{j\in[n]\setminus i}$ are receivers. We will also use $\mathsf{msg}_{\Psi}^{(i)}$, to indicate a message for Protocol Ψ generated by Party P_i.

We now define three relations that will be used in the protocol, and we define languages $\mathcal{L}_\alpha = \{x : \exists w \text{ such that } \mathcal{R}_\alpha(x,w)=1\}$ for $\alpha \in \{\mathsf{NIWI}_1, \mathsf{NIWI}_2, \mathsf{mCDS}\}$.

- $\mathcal{R}_{\mathsf{NIWI}_1}\Big((c_1,c_2),r\Big) = 1 \iff \Big(c_1 = \mathsf{NMCom}_{\mathsf{tag=0}}(0;r) \bigvee c_2 = \mathsf{NMCom}_{\mathsf{tag=0}}(0;r)\Big)$
- $\mathcal{R}_{\mathsf{NIWI}_2}\Big((m_1, m_2, \{m_3^k\}_{k\in[n]}, \{\mathsf{stmt}^k_{\mathsf{mCDS}}\}_{k\in[n]}, com, x, M, c_1, j, \{c_y^k, c_z^k\}_{k\in[n]}),$ $(\mathsf{st}, w_x, w_r, r, \{\widehat{r_k}\}_{k\in[n]}, \{\widetilde{r}^k\}_{k\in[n]})\Big) = 1 \iff$ $\Big((m_1,\mathsf{st}) = \mathsf{smMPC}(w_x; w_r) \bigwedge m_2 = \mathsf{mCDS.E}(com, x, \mathsf{smMPC}(M, \mathsf{st}; w_r)) \bigwedge$ $\forall k \in [n], m_3^k = \mathsf{mCDS.Prove}(\widetilde{\mathsf{st}}, \mathsf{stmt}^k_{\mathsf{mCDS}})$ where $(\widetilde{m}, \widetilde{st}) = \mathsf{mCDS.Com}(1^{\kappa_{\mathsf{mCDS.R}}}, (w_x, w_r, 0^{n\lambda}), j; \widetilde{r}^k)\Big)$ $\bigvee \Big(c_1 = \mathsf{NMCom}_{\mathsf{tag=j}}(\widehat{r_1}||\ldots||\widehat{r_n}; r) \wedge \forall k \in [n], \big(c_y^k = \mathsf{NMCom}_{\mathsf{tag=0}}(0;\widehat{r_k}) \vee$ $c_z^k = \mathsf{NMCom}_{\mathsf{tag=0}}(0;\widehat{r_k})\big)\Big)$
- $\mathcal{R}_{\mathsf{mCDS}}\Big((m_1, c_1, j, \{c_y^k, c_z^k\}_{k\in[n]}), (w, r, \{\widehat{r_k}\}_{k\in[n]})\Big) = 1 \iff m_1 = \mathsf{smMPC}(w;r) \bigvee$ $\Big(c_1 = \mathsf{NMCom}_{\mathsf{tag=j}}(\widehat{r_1}||\ldots||\widehat{r_n}; r) \wedge \forall k \in [n], \big(c_y^k = \mathsf{NMCom}_{\mathsf{tag=0}}(0;\widehat{r_k}) \vee c_z^k =$ $\mathsf{NMCom}_{\mathsf{tag=0}}(0;\widehat{r_k})\big)\Big)$

In words, $\mathcal{R}_{\mathsf{NIWI}_1}$ is stating that one of two non-malleable commitments is to 0. $\mathcal{R}_{\mathsf{NIWI}_2}$ is stating that either i) first and second round of the semi-malicious MPC are computed correctly, and the MCDS commitment and proofs are computed correctly OR ii) the trapdoor is known. $\mathcal{R}_{\mathsf{mCDS}}$ is stating that either i) the first round of the semi-malicious MPC is computed correctly OR ii) the trapdoor is known. In Fig. 1, Fig. 2 and Fig. 3, we describe the construction of our two round maliciously-secure MPC protocol fmMPC. We have the following theorem.

Theorem 9. *Fix an arbitrary polynomial $n = n(\lambda)$ for security parameter λ. Assuming sub-exponentially secure NIWI proofs satisfying Definition 3, n-special non-malleable commitments satisfying Definition 7, sub-exponentially secure MPC against semi-malicious adversaries according to Definition* **??** *and subexponentially secure multi-party CDS according to Definitions 10 and 11, two round*

Protocol fmPMC - Round 1

Common input: Security parameter 1^λ and number of parties 1^n

P_i **input**: $x_i \in \{0,1\}^{p(\lambda)}$

Round 1: For $i \in [n]$, P_i computes the following.

- $(\mathsf{msg1}^{(i)}_{\mathsf{smMPC}}, \mathsf{st}^{(i)}_{\mathsf{smMPC}}) = \mathsf{smMPC}(x_i; r_i)$, the first semi-malicious MPC protocol message with input x_i, randomness r_i.
- $\mathsf{cmt}^{(i)}_{\mathsf{td}} = \mathsf{NMCom}_{\mathsf{tag=i}}(0^{n\lambda}; r_0)$, a non-malleable commitment to $0^{n\lambda}$.
- $\mathsf{cmt}^{(i)}_{y} = \mathsf{NMCom}_{\mathsf{tag=0}}(0; r_y)$, a non-malleable commitment to the bit 0 with randomness r_y.
- $\mathsf{cmt}^{(i)}_{z} = \mathsf{NMCom}_{\mathsf{tag=0}}(0; r_z)$, a non-malleable commitment to the bit 0 with randomness r_z.
- $\pi^{(i)}_{\mathsf{NIWI}_1} \leftarrow \mathsf{NIWIProve}(x_{\mathsf{NIWI}_1}, w_{\mathsf{NIWI}_1}, \mathcal{L}_{\mathsf{NIWI}_1})$ where $x_{\mathsf{NIWI}_1} = (\mathsf{cmt}^{(i)}_y, \mathsf{cmt}^{(i)}_z)$ and $w_{\mathsf{NIWI}_1} = r_y$.
- For all $j \in [n] \setminus i$, compute an mCDS commitment

$$(\mathsf{msg1}^{(i)}_{\mathsf{mCDS}^{(j)}}, \mathsf{st}^{(i)}_{\mathsf{mCDS}^{(j)}}) = \mathsf{mCDS.Com}(1^{\kappa_{\mathsf{mCDS.R}}}, (x_i, r_i, 0^{n\lambda}), i; r^{(i)}_{\mathsf{mCDS}^{(j)}}).$$

  Here $\kappa_{\mathsf{mCDS.R}}$ indicates the receiver security parameter of the mCDS.

For $i \in [n]$, P_i broadcasts

$$\left(\mathsf{msg1}^{(i)}_{\mathsf{smMPC}}, \mathsf{cmt}^{(i)}_{\mathsf{td}}, \mathsf{cmt}^{(i)}_{y}, \mathsf{cmt}^{(i)}_{z}, \pi^{(i)}_{\mathsf{NIWI}_1}\right),$$

and sends $\mathsf{msg1}^{(i)}_{\mathsf{mCDS}^{(j)}}$ to P_j for each $j \in [n], j \neq i$.

For $i, j \in [n], j \neq i$, P_i receives from P_j

$$\left(\mathsf{msg1}^{(j)}_{\mathsf{smMPC}}, \mathsf{cmt}^{(j)}_{\mathsf{td}}, \mathsf{cmt}^{(j)}_{y}, \mathsf{cmt}^{(j)}_{z}, \pi^{(j)}_{\mathsf{NIWI}_1}, \mathsf{msg1}^{(j)}_{\mathsf{mCDS}^{(i)}}\right)$$

For $i \in [n]$, P_i verifies each $\pi^{(j)}_{\mathsf{NIWI}_1}$, and outputs Abort if verification fails.

Fig. 1: Round 1 of a two-round maliciously secure MPC protocol

maliciously-secure MPC for n-parties with super-polynomial simulation exists which satisfies Definition **??**.

Proof. In what follows, we let $\delta = 2^{-n\gamma}$ where γ denotes the size of randomness r_y (and equivalently r_z) in the protocol, and n denotes the number of parties which is polynomial in λ.

- We will rely on any NMCom that is a n-special one-to-one non-malleable commitment, according to Definition 7. We use $T^{\mathsf{brk}}_{\mathsf{NMCom}}$ to denote the time needed to extract the committed bit from any commitment string via a brute-force attack.

Protocol fmMPC **- Round 2**

Round 2: For $i \in [n]$, P_i computes the following.

- Compute the second semi-malicious MPC protocol message

$$(\mathsf{msg2}^{(i)}_{\mathsf{smMPC}}, \mathsf{st}^{(i)}_{\mathsf{smMPC}}{}') = \mathsf{smMPC}(\{\mathsf{msg1}^{(k)}_{\mathsf{smMPC}}\}_{k\in[n]}, \mathsf{st}^{(i)}_{\mathsf{smMPC}}; r_i).$$

- Compute the mCDS encryption

$$\mathsf{msg2s}_{\mathsf{mCDS}^{(i)}} \leftarrow \mathsf{mCDS.E}(\{\mathsf{msg1}^{(j)}_{\mathsf{mCDS}^{(i)}}\}_{j\in[n]\setminus i}, \{x^{(j)}_{\mathsf{mCDS}}\}_{j\in[n]\setminus i}, \mathsf{msg2}^{(i)}_{\mathsf{smMPC}}),$$

  where
$$x^{(j)}_{\mathsf{mCDS}} = (\mathsf{msg1}^{(j)}_{\mathsf{smMPC}}, \mathsf{cmt}^{(j)}_{\mathsf{td}}, j, \{\mathsf{cmt}^{(k)}_y, \mathsf{cmt}^{(k)}_z\}_{k\in[n]}).$$
- For $j \in [n] \setminus i$, compute the mCDS proof

$$\mathsf{msg2r}^{(i)}_{\mathsf{mCDS}^{(j)}} \leftarrow \mathsf{mCDS.Prove}(\mathsf{st}^{(i)}_{\mathsf{mCDS}^{(j)}}, x^{(i)}_{\mathsf{mCDS}}),$$

  where
$$x^{(i)}_{\mathsf{mCDS}} = (\mathsf{msg1}^{(i)}_{\mathsf{smMPC}}, \mathsf{cmt}^{(i)}_{\mathsf{td}}, i, \{\mathsf{cmt}^{(k)}_y, \mathsf{cmt}^{(k)}_z\}_{k\in[n]}).$$
- Compute NIWI proof

$$\pi_{\mathsf{NIWI}_2}^{(i)} \leftarrow \mathsf{NIWIProve}(x_{\mathsf{NIWI}_2}, w_{\mathsf{NIWI}_2}, \mathcal{L}_{\mathsf{NIWI}_2}),$$

  where

$$x_{\mathsf{NIWI}_2} = \begin{pmatrix} \mathsf{msg1}^{(i)}_{\mathsf{smMPC}}, \mathsf{msg2s}_{\mathsf{mCDS}^{(i)}}, \{\mathsf{msg2r}^{(i)}_{\mathsf{mCDS}^{(j)}}\}_{j\in[n]}, \{x^{(k)}_{\mathsf{mCDS}}\}_{k\in[n]}, \\ \{\mathsf{msg1}^{(j)}_{\mathsf{mCDS}^{(i)}}\}_{j\in[n]\setminus i}, \{x^{(j)}_{\mathsf{mCDS}}\}_{j\in[n]\setminus i}, \\ \{\mathsf{msg1}^{(k)}_{\mathsf{smMPC}}\}_{k\in[n]}, \mathsf{cmt}^{(i)}_{\mathsf{td}}, i, \{\mathsf{cmt}^{(k)}_y, \mathsf{cmt}^{(k)}_z\}_{k\in[n]} \end{pmatrix}$$

  and $w_{\mathsf{NIWI}_2} = \left(\mathsf{st}^{(i)}_{\mathsf{smMPC}}, x_i, r_i, 0, 0^*, \{r^{(i)}_{\mathsf{mCDS}^{(j)}}\}_{j\in[n]}\right)$.

For $i \in [n]$, P_i broadcasts

$$\left(\mathsf{msg2s}_{\mathsf{mCDS}^{(i)}}, \left\{\mathsf{msg2r}^{(i)}_{\mathsf{mCDS}^{(j)}}\right\}_{j\in[n]\setminus i}, \pi^{(i)}_{\mathsf{NIWI}_2}\right).$$

For $i \in [n]$, P_i receives

$$\left(\{\mathsf{msg2s}_{\mathsf{mCDS}^{(j)}}\}_{j\in[n]\setminus i}, \left\{\mathsf{msg2r}^{(k)}_{\mathsf{mCDS}^{(j)}}\right\}_{k\in[n]\setminus i, j\in[n]\setminus i}, \left\{\pi^{(j)}_{\mathsf{NIWI}_2}\right\}_{j\in[n]\setminus i}\right).$$

Fig. 2: Round 2 of a two-round maliciously secure MPC protocol

- We also rely on any NIWI_1 that is $(T^{\mathsf{brk}}_{\mathsf{NMCom}}, \lambda)$-secure, any MCDS that satisfies $(T^{\mathsf{brk}}_{\mathsf{NMCom}}, 1/\delta)$ receiver security and $(\lambda, 1/\delta)$ sender security.
- We will rely on any semi-malicious MPC that is $(\max(T^{\mathsf{brk}}_{\mathsf{NMCom}}, T^{\mathsf{brk}}_{\mathsf{mCDS}}), 1/\delta)$ secure.
- Will rely on (standard) polynomial-size hardness of NIWI_2.

Protocol fmMPC **- Output Reconstruction**

Output Reconstruction: P_i computes the following.

– Verify each $\pi^{(j)}_{\mathsf{NIWI}_2}$ and output Abort if verification fails.
– For all $j \in [n]$, reconstruct

$$\mathsf{msg2}^{(i)}_{\mathsf{smMPC}} \leftarrow \mathsf{mCDS.Rec}(\mathsf{msg2s}_{\mathsf{mCDS}^{(j)}}, \{\mathsf{msg2}^{(k)}_{\mathsf{mCDS}^{(j)}}\}_{k\in[n]\setminus j}).$$

– Use $\{\mathsf{msg2}^{(j)}_{\mathsf{smMPC}}\}_{j\in[n]}$ and $\mathsf{st}^{(i)}_{\mathsf{smMPC}}{}'$ to compute the output of the smMPC.

Fig. 3: Output reconstruction for a two-round maliciously secure MPC protocol

Towards the end of the proof, we discuss how to set security parameters of these primitives (assuming subexponential security of all primitives) to achieve all relationships discussed above.

We will now describe the simulator for fmMPC protocol. Below, H is the set of honest parties and M is the set of malicious parties (i.e. parties corrupted by the adversary ADV):

$\mathsf{Sim}_{\mathsf{fmMPC}}$:

– **Simulation of Round 1**: For all $i \in H$:
  - Guess the randomness used in $\mathsf{cmt}^{(j)}_y$ or $\mathsf{cmt}^{(j)}_z$ for all $j \in [n]$. Let the guessed values be $\{v'_1, \ldots, v'_n\}$. Use these guessed values to generate $\mathsf{cmt}^{(i)}_{\mathsf{td}}$ using randomness r'_i sampled uniformly at random.
  - Generate $\mathsf{msg1}^{(i)}_{\mathsf{smMPC}}$ using $\mathsf{Sim}_{\mathsf{smMPC}}$
  - Generate $\mathsf{cmt}^{(i)}_y$, $\mathsf{cmt}^{(i)}_z$ and $\pi^{(i)}_{\mathsf{NIWI}}$ as per the honest fmMPC protocol
  - For all $j \in [n]$, use the message and randomness for $\mathsf{cmt}^{(i)}_{\mathsf{td}}$ to generate $\mathsf{msg1}^{(i)}_{\mathsf{mCDS}^{(j)}} \leftarrow \mathsf{mCDS.Com}(1^{\kappa_{\mathsf{mCDS.R}}}, (0, r'_i, v'_1, \ldots, v'_n), i)$.
  - Send the generated items as prescribed in the honest fmMPC protocol, receive items from all parties P_j where $j \in M$ and Abort if any of the $\pi^{(j)}_{\mathsf{NIWI}}$ is invalid.

– **Checking the guess correctness**: Perform the following $\varnothing$-Check: For every $j \in M$, if $\mathsf{cmt}^{(j)}_y = \mathsf{NMCom}_{\mathsf{tag}=0}(0; v'_j)$ or $\mathsf{cmt}^{(j)}_z = \mathsf{NMCom}_{\mathsf{tag}=0}(0; v'_j)$, the check passes and the simulation proceeds to Round 2. Otherwise, the check fails and the simulation goes back to Round 1

– **Extracting the** mCDS **inputs**: For all $j \in M$, use brute-force to break their MCDS receiver messages $\{\mathsf{msg1}^{(j)}_{\mathsf{mCDS}^{(i)}}\}_{i\in[n]\setminus\{j\}}$. If input extraction succeeds, i.e., if for every $j \in M$, there exists $i \in [n] \setminus \{j\}$, $(x_j, r_j), r^{(j)}_{\mathsf{mCDS}^{(i)}}$ such that

$$\mathsf{msg1}^{(j)}_{\mathsf{mCDS}^{(i)}} = \mathsf{mCDS.Com}(1^{\kappa_{\mathsf{mCDS.R}}}, (x_j, r_j, 0^{n\lambda}), j; r^{(j)}_{\mathsf{mCDS}^{(i)}}),$$

then send $(x_j, r_j)_{j\in M}$ to $\mathsf{Sim}_{\mathsf{smMPC}}$ and obtain $\mathsf{msg2}^{(i)}_{\mathsf{smMPC}}$ for $i \in H$ from $\mathsf{Sim}_{\mathsf{smMPC}}$. If input extraction fails, set $\mathsf{msg2}^{(i)}_{\mathsf{smMPC}}$ for $i \in H$ to $0^{s(\lambda)}$, where $s(\lambda)$ denotes the length of round 2 semi-malicious MPC messages.

- **Simulation of Round 2**: For all $i \in H$:
  - Generate $\mathsf{msg2s}_{\mathsf{mCDS}^{(i)}}$ as per the honest fmMPC protocol.
  - For all $j \in [n] \setminus i$, generate the mCDS proof as per the honest fmMPC protocol.
  - Generate NIWI proof $\pi^{(i)}_{\mathsf{NIWI}_2}$ using $(0, 0, 0, r'_i, \{v'_1, \ldots, v'_n\}, 0^*)$ as the witness w_{NIWI_2}
  - Send the generated items as prescribed in the honest fmMPC protocol.
- **Output Reconstruction**: Receive items from all parties P_j where $j \in M$, and perform the first two steps of Output Reconstruction as prescribed in the honest fmMPC protocol. Finally, send $\{\mathsf{msg2}^{(j)}_{\mathsf{smMPC}}\}_{i \in M}$ to $\mathsf{Sim}_{\mathsf{smMPC}}$.

In Appendix **??** in the full version, we describe a sequence of hybrids, transitioning from the real world to the ideal world and prove, via a sequence of lemmas, that these hybrids are indistinguishable from each other, thus proving that our protocol fmMPC satisfies Theorem 9.

4.1 Compactness and Reusability

We sketch modification to our protocol to achieve communication complexity independent of the circuit size (compactness) and to allow parties to reuse the first message to compute unbounded, but polynomially many, functions (reusability).

Compactness. Instantiating the semi-malicious MPC with a compact protocol [AJJM20] results in a compact malicious MPC, except for the NIWI used in the second round that is used to prove a statement related to the semi-malicious MPC, and therefore may be non-compact. However we note that we can generically transform any non-compact NIWI into a compact one using (perfectly correct) fully-homomorphic encryption (FHE). The transformation is analogous to [GGI+15] and we outline it here for completeness.

The NIWI prover samples two FHE key pair $(\mathsf{sk}_0, \mathsf{pk}_0)$ and $(\mathsf{sk}_1, \mathsf{pk}_1)$ and compute two encryptions of the witness $c_0 = \mathsf{FHE.Enc}(\mathsf{pk}_0, w)$ and $c_1 = \mathsf{FHE.Enc}(\mathsf{pk}_1, w)$. Then it homorphically computes the predicate $\mathcal{R}(\cdot, x)$ to obtain two evaluated ciphertexts e_0 and e_1. Finally, it computes a NIWI proof that EITHER $(\mathsf{sk}_0, \mathsf{pk}_0)$ and c_0 are well-formed and e_0 is an encryption of 1 OR $(\mathsf{sk}_1, \mathsf{pk}_1)$ and c_1 are well-formed and e_1 is an encryption of 1. The verifier simply checks that the NIWI correctly verifies and that e_0 and e_1 are the correct output of the evaluation algorithm for the circuit $\mathcal{R}(\cdot, x)$. One can show with a standard argument that the proof is still witness indistinguishable. Furthermore, the communication complexity does only depend polynomially on $|w|$, by compactness of the FHE.

Reusable First Message. Instantiating a semi-malicious MPC with one with reusable first message [AJJM20,BGMM20,BL20] and the reusable variant of our MCDS, we obtain 2-round malicious MPC where the first message can be reused an unbounded amount of times (possibly to compute different functions).

References

[ACJ17] Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A new approach to round-optimal secure multiparty computation. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 468–499. Springer, Heidelberg, August 2017.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004.

[AIK05] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications (extended abstract). In *20th Annual IEEE Conference on Computational Complexity (CCC'05)*, pages 260–274, 2005.

[AJJM20] Prabhanjan Ananth, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Multikey fhe in the plain model. *IACR ePrint Arch.*, 2020:180, 2020.

[BGI+17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 275–303. Springer, Heidelberg, December 2017.

[BGJ+17] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Dakshita Khurana, and Amit Sahai. Round optimal concurrent MPC via strong simulation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 743–775. Springer, Heidelberg, November 2017.

[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 Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 459–487. Springer, Heidelberg, August 2018.

[BGMM20] James Bartusek, Sanjam Garg, Daniel Masny, and Pratyay Mukherjee. Reusable two-round mpc from ddh. Cryptology ePrint Archive, Report 2020/170, 2020.

[BHP17] Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 645–677. Springer, Heidelberg, November 2017.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012*, pages 784–796. ACM Press, October 2012.

[BJKL21] Fabrice Benhamouda, Aayush Jain, Ilan Komargodski, and Huijia Lin. Multiparty reusable non-interactive secure computation from lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 724–753. Springer, 2021.

[BL18a] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 500–532. Springer, Heidelberg, April / May 2018.

[BL18b] Nir Bitansky and Huijia Lin. One-message zero knowledge and nonmalleable commitments. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018,* *Panaji, India, November 11-14, 2018, Proceedings, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 209–234. Springer, 2018.

[BL20] Fabrice Benhamouda and Huijia Lin. Multiparty reusable non-interactive secure computation. Cryptology ePrint Archive, Report 2020/221, 2020.

[BP14] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 353–370. Springer, Heidelberg, August 2014.

[CCG+19] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. Cryptology ePrint Archive, Report 2019/216, 2019.

[GG14] Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 477–495. Springer, Heidelberg, May 2014.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 74–94. Springer, Heidelberg, February 2014.

[GGI+15] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, October 2015.

[GIKM98] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *30th ACM STOC*, pages 151–160. ACM Press, May 1998.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.*, 7(1):1–32, 1994.

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006.

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Heidelberg, May / June 2006.

[GP15] Sanjam Garg and Antigoni Polychroniadou. Two-round adaptively secure MPC from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 614–637. Springer, Heidelberg, March 2015.

[GRS+11] Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648. Springer, Heidelberg, August 2011.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 468–499. Springer, Heidelberg, April / May 2018.

[HHPV18] Shai Halevi, Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Round-optimal secure multi-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 488–520. Springer, Heidelberg, August 2018.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Heidelberg, August 2017.

[KK19] Yael Tauman Kalai and Dakshita Khurana. Non-interactive non-malleability from quantum supremacy. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 552–582. Springer, Heidelberg, August 2019.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 335–354. Springer, Heidelberg, August 2004.

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of Yao's protocol for two-party computation. *Journal of Cryptology*, 22(2):161–188, April 2009.

[LPS17] H. Lin, R. Pass, and P. Soni. Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In *2017 IEEE 58th Annual Symposium (FOCS)*, pages 576–587, 2017.

[MPP20] Andrew Morgan, Rafael Pass, and Antigoni Polychroniadou. Succinct non-interactive secure computation. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 216–245. Springer, Heidelberg, May 2020.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2003.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 57–74. Springer, Heidelberg, August 2008.

[PS04] Manoj Prabhakaran and Amit Sahai. New notions of security: Achieving universal composability without trusted setup. In László Babai, editor, *36th ACM STOC*, pages 242–251. ACM Press, June 2004.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.