Concurrent-Secure Two-Party Computation in Two Rounds from Subexponential LWE

Saikrishna Badrinarayanan[*] Rex Fernando[†] Amit Sahai[‡]

November 2, 2021

Abstract

Very recently, two works were able to construct two-round secure multi-party computation (MPC) protocols in the plain model, without setup, relying on the super-polynomial simulation framework of Pass [Pas03]. The first work [ABG$^+$21] achieves this relying on subexponential non-interactive witness indistinguishable arguments, the subexponential SXDH assumption, and the existence of a special type of non-interactive non-malleable commitment. The second work [FJK21] additionally achieves concurrent security, and relies on subexponential quantum hardness of the learning-with-errors (LWE) problem, subexponential classical hardness of SXDH, the existence of a subexponentially-secure (classically-hard) indistinguishablity obfuscation (iO) scheme, and time-lock puzzles.

This paper focuses on the assumptions necessary to construct secure computation protocols in two rounds without setup, focusing on the subcase of two-party functionalities. In this particular case, we show how to build a two-round, *concurrent-secure*, two-party computation (2PC) protocol *based on a single, standard, post-quantum* assumption, namely subexponential hardness of the learning-with-errors (LWE) problem.

We note that our protocol is the first two-round concurrent-secure 2PC protocol that does not require the existence of a one-round non-malleable commitment (NMC). Instead, we are able to use the two-round NMCs of [KS17a], which is instantiable from subexponential LWE.

1 Introduction

Secure computation is a fundamental primitive in cryptography which allows two or more parties, all of whom have private inputs, to collectively compute some function over their inputs securely without revealing the inputs themselves. In recent years, significant attention has been devoted to the round-complexity of secure computation in the setting of two parties, as well as in the multi-party setting (MPC). This has culminated in recent work

[*] VISA Research. Work done while at UCLA. Email: `bsaikrishna7393gmail.com`.

[†] UCLA. Email: `rex1fernandogmail.com`.

[‡] UCLA. Email: `sahaics.ucla.edu`.

of [GMPP16, ACJ17, BHP17, BGJ$^+$18, HHPV18, CCG$^+$20], which give protocols that run in four rounds, known to be the least amount of rounds possible for full security in the plain model[1]

The results above achieve security in the *standalone* setting, where all parties are assumed to participate in only one instance of the protocol. A more realistic setting allows parties to participate *concurrently* in arbitrarily many instances. Unfortunately, Barak, Prabhakaran and Sahai [BPS06] show that achieving the standard definition of concurrent security is impossible *in any rounds* in the plain model, without a trusted setup.

One standard relaxation of simulation security, which is widely used to circumvent many lower-bound results, is the notion of *super-polynomial simulation*, or SPS [Pas03]. With this notion, for any real-world adversary, we require an ideal-world simulator which runs in super-polynomial time. In 2017, the work of Badrinarayanan et al. [BGJ$^+$17] used this notion to circumvent both the impossibility of concurrent MPC and the four-round lower-bound, giving a protocol which works in three rounds and satisfies concurrent security. For several years, this result was the best-known result regarding the round-complexity of MPC in the plain model. Until very recently, no two-round protocols were known. This was the case even in in the restricted setting of two-party computation (where both parties receive output).

Earlier this year, two new works were published, both of which improve the state of the art in this area:

- The work of [ABG$^+$21] gave a two-round MPC protocol for general functionalities which achieves standalone security in the plain model without setup and with a super-polynomial simulator, assuming subexponential non-interactive witness-indistinguishable arguments, the subexponential SXDH assumption, and the existence of a special type of non-interactive non-malleable commitment.[2]

- The work of [FJK21] gave a concurrent, highly-reusable[3] two-round MPC protocol for general functionalities, assuming subexponential quantum hardness of the learning-with-errors (LWE) problem, subexponential classical hardness of SXDH, the existence of a subexponentially-secure (classically-hard) indistinguishablity obfuscation (iO) scheme, and time-lock puzzles.

Assumptions for two-party secure computation. The goal of our work is to focus on secure computation in the two-party setting, and to explore the assumptions under which two-round secure protocols are possible. Even in this more specific setting, the two above results are the only known protocols which achieve two-round protocols for general two-party functionalities.[4] Both of the previously mentioned works on two-round protocols use

[1] It is also known how to achieve two-round MPC that satisfies a much weaker notion of *semi-malicious* security, where the adversary is assumed to follow the honest protocol specification. [GS17] Alternately, achieving full security in two rounds is possible if we allow for a trusted setup. In this paper, we focus on achieving full malicious security in the plain model, without setup.

[2] The protocol of [ABG$^+$21] is given in the form of a compiler which transforms a two-round semi-malicious-secure MPC protocol into a malicious-secure one.

[3] See [FJK21] for the exact definition of reusability obtained.

[4] If we restrict ourselves to functionalities where only one party receives output, then it is known how to achieve two-round secure computation from much simpler assumptions [BGI$^+$17], in the setting of standalone security.

powerful primitives which are only known from strong assumptions. More specifically, the work of [ABG$^+$21] requires a strong version of non-interactive non-malleable commitments, which are only known from strong, non-standard assumptions, such as adaptive one-way functions [PPV08], or keyless hash functions along with a subexponential variant of the "hardness amplifiability" assumption of [BL18]. The work of [FJK21] is able to avoid using these strong commitments, instead using (a modified version of) the one-round NMC of [Khu21], which relies on the existence of sub-exponential indistinguishability obfuscation (iO).

We briefly discuss the assumptions under which iO exists. Our understanding of these assumptions has vastly improved in recent years, culminating in the work of [JLS21b, JLS21a], which showed that iO can be built on well-founded assumptions, namely hardness of LPN over $\mathbb{F}_p$, hardness of DLIN, and the existence of PRGs in NC^0. However, our understanding of the assumptions necessary for *quantum-secure* iO is much less stable. We note that besides the above mentioned work, all other constructions of iO rely on ad-hoc hardness assumptions which were specifically invented for the purpose of proving security of iO [GGH$^+$13, BGK$^+$14, BR14, PST14, AGIS14, BMSZ16, CLT13, CLT15, GGH15, CHL$^+$15, BWZ14, CGH$^+$15, HJ15, BGH$^+$15, Hal15, CLR15, MF15, MSZ16, DGG$^+$16, Lin16, LV16, AS17, Lin17, LT17, GJK18, AJS18, Agr19, LM18, JLMS19, BIJ$^+$20, AP20, BDGM20a, GP20, BDGM20b, WW20]. Although some of the most recent of these constructions rely on lattice-based assumptions which ostensibly could be quantum-secure [GP20, BDGM20b, WW20], there are already preliminary attacks on some versions of these new assumptions [HJL21], and thus it is unclear whether the constructions are secure.

In addition to iO, both of the constructions of two-round MPC above use other assumptions (i.e, SXDH) which, while standard, are quantum-broken. With all of this in mind, it is interesting to ask whether it is possible to construct two-round protocols using simpler assumptions, especially post-quantum ones. As mentioned above, this question is interesting even if we restrict ourselves to the case of two parties, since up to this point the only known results even in this subcase are the two discussed above.

1.1 Our Results

In this work, we make significant process in answering the above question, focusing. In this particular case, we show how to build a two-round, *concurrent-secure*, two-party secure computation protocol *based on a single, standard, post-quantum* assumption, namely subexponential hardness of the learning-with-errors (LWE) problem. We state our main theorem now.

Theorem 1. *Assuming sub-exponential hardness of the learning-with-errors (LWE) problem, there exists a two-round two-party computation protocol for any polynomial-time functionality f where both parties receive outputs, in the plain model with super-polynomial simulation.*

We note that our protocol is the first two-round concurrent-secure 2PC protocol that does not require the existence of a one-round non-malleable commitment (NMC). Instead, we are able to use the two-round NMCs of [KS17a], which is instantiable from subexponential LWE. Our protocol is also the first such protocol that does not require the existence of non-interactive witness indistinguishable arguments or time-lock puzzles.

1.2 Concurrent Work

Concurrently and independently to our result, the work of [AMR21] also achieves a two-round concurrent-secure 2PC protocol, assuming subexponential hardness of LWE *and* subexponentially-secure non-interactive non-malleable commitments, which are only known under non-standard assumptions.

The authors of [AMR21] also propose several applications. One particularly interesting application involves plugging the protocol into the work of Bartusek et al. [BCKM21] to get the first concurrent-secure quantum 2PC in the plain model. To achieve this result, the protocol of [AMR21] must be assumed to be quantum-secure. That is, there must exist a quantum-secure construction of non-interactive NMCs. Currently the best-known assumption under which non-interactive NMCs can be constructed is the existence of indistinguishability obfuscation (iO) [Khu21]. As discussed above, our understanding of post-quantum iO is very limited, and there are no constructions based on well-established assumptions. In contrast, our protocol only relies on the learning-with-errors assumption, and thus we achieve the same application as [AMR21], relying only on one standard post-quantum assumption (subexponentially secure LWE).

2 Preliminaries

In the following, we write $T_1 \ll T_2$ for functions T_1 and T_2 if for all polynomials p, $p(T_1(\lambda))$ is asymptotically smaller than $T_2(\lambda)$. We denote with $\mathcal{G}(x;r)$ the execution of a probabilistic algorithm $\mathcal{G}$, where x is the input to the algorithm and r is the string of random coins. When we do not need to explicitly deal with the random coins of $\mathcal{G}$, we write $\mathcal{G}(x)$ and assume that the coins r are chosen uniformly at random.

2.1 Two-Round SPS Strong Zero Knowledge

We define the notion of two-round strong zero knowledge with super-polynomial simulation first given in [KS17a]. Here *strong* means that the zero-knowledge property holds even against adversaries which themselves are strong enough to run the simulator.

We consider zero-knowledge protocols with the following syntax. All algorithms below are polynomial-time.

- $\mathsf{ZK}_1(1^\lambda; r) \to \mathbf{zk}_1$ takes as input the security parameter 1^λ along with randomness r and produces the verifier's message.
- $\mathsf{ZK}_2(1^\lambda, \phi, w, \mathbf{zk}_1; r') \to \mathbf{zk}_2$ takes as input security parameter, the statement ϕ and the witness w along with the verifier's message and randomness r' and produces the prover's message.
- $\mathsf{ZK}_{\mathsf{verify}}(\phi, \mathbf{zk}_2, r)$ is a deterministic algorithm which takes the statement τ along with the prover's message and the randomness used to generate the verifier's message and accepts or rejects.

Definition 1 (($T_{\mathsf{sound}}, T_{\mathsf{Sim}}, T_{\mathbf{zk}}, T_L, \epsilon_{\mathsf{sound}}, \epsilon_{\mathbf{zk}}$)-SPSS Zero-Knowledge Arguments)**.** *Let L be a language in NP which is decidable in time T_L, with a polynomial-time computable relation R_L. Let T_{sound}, T_{Sim}, $T_{\mathbf{zk}}$ be superpolynomial functions and $\epsilon_{\mathbf{zk}}$, $\epsilon_{\mathsf{sound}}$ negligible functions, where $T_{\mathsf{sound}} \ll T_{\mathsf{Sim}} \ll T_{\mathbf{zk}} \ll T_L$. A protocol between P and V is a ($T_{\mathsf{sound}}, T_{\mathsf{Sim}}, T_{\mathbf{zk}}, T_L$, $\epsilon_{\mathsf{sound}}, \epsilon_{\mathbf{zk}}$)-strong zero-knowledge argument for L if it satisfies the following properties:*

- ***Perfect Completeness.*** *For every security parameter 1^λ and $(x, w) \in R_L$, it holds that*

$$\Pr\left[\langle P(w), V\rangle(1^\lambda, x) = 1\right] = 1,$$

where the probability is over the random coins of P and V.

- ($T_{\mathsf{sound}}, \epsilon_{\mathsf{sound}}$)-***Adaptive Soundness.*** *For every polynomial $p(\lambda)$ and every prover $P^* \in T_{\mathsf{sound}}$ that given 1^λ and an honest verifier message $\mathbf{zk}_1$, chooses an input length 1^p for some polynomial $p \in \mathsf{poly}(\lambda)$, and then chooses $x \in \{0,1\}^p \setminus L$ and outputs $(x, \mathbf{zk}_2)$, it holds that*

$$\Pr\left[\mathsf{ZK}_{\mathsf{verify}}(x, \mathbf{zk}_2, r) = 1\right] \leq \epsilon_{\mathsf{sound}}(\lambda),$$

where r is the randomness used to generate zk_1 and the probability is over the random coins of V.

- ($T_{\mathsf{Sim}}, T_{\mathbf{zk}}, \epsilon_{\mathbf{zk}}$)***-Strong Zero-Knowledge.*** *There exists a (uniform) simulator Sim which runs in time T_{Sim} which takes as input the round-one transcript $\mathbf{zk}_1$ and a statement x such that the following holds. Consider an adversary V^* which runs in time $T_{\mathbf{zk}}$ that takes as input 1^λ and advice z and outputs a verifier's first round message $\mathbf{zk}_1^*$. Then, for all $(x, w) \in R_L$, distinguishers $\mathcal{D}$ which run in time $T_{\mathbf{zk}}$, and advice z,*

$$\left|\Pr\left[\mathcal{D}(x, z, r, \mathsf{ZK}_2(1^\lambda, x, w, \mathbf{zk}_1^*)) = 1\right] - \Pr\left[\mathcal{D}(x, z, r, \mathsf{Sim}(1^\lambda, x, \mathbf{zk}_1^*)) = 1\right]\right| < \epsilon_{\mathbf{zk}}(\lambda),$$

where r is the private randomness of V^*.

2.2 Two-Round Statistically-Sender-Private Oblivious Transfer

We give the formal definition of two-round oblivious transfer, where the receiver's security is computational, and there exists a (possibly computationally unbounded) extractor for the receiver's first-round message such that statistical security holds for the sender. The OT scheme consists of the following polynomial-time algorithms:

- $\mathsf{OT}_1(1^\lambda, b; r) \to \mathbf{ot}_1$: The receiver's OT_1 algorithm takes a choice bit b and produces the receiver's OT message.

- $\mathsf{OT}_2(1^\lambda, \ell_0, \ell_1, \mathbf{ot}_1; r') \to \mathbf{ot}_2$: The sender's OT_2 algorithm takes a pair of strings to choose from along with the receiver's OT message and produces the sender's OT message.

- $\mathsf{OT}_3(\mathbf{ot}_2; r) \to \ell_b$: The receiver's OT_3 takes the sender's OT message and outputs ℓ_b.

Definition 2. *A tuple* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *is a* $(T_R, \epsilon_R, \epsilon_S)$*-statistically-sender-private oblivious transfer algorithm if the following properties hold:*

- ***Correctness.*** *For all* λ, b, ℓ_0, ℓ_1,

$$\Pr\left[\mathsf{OT}_3(\mathbf{ot}_2; r) = \ell_b \middle| \begin{array}{l} \mathbf{ot}_1 \leftarrow \mathsf{OT}_1(1^\lambda, b; r) \\ \mathbf{ot}_2 \leftarrow \mathsf{OT}_2(1^\lambda, \ell_0, \ell_1, \mathbf{ot}_1) \end{array}\right] = 1.$$

- (T_R, ϵ_R)***-Computational Receiver Privacy.*** *For all machines* $\mathcal{D}$ *running in time at most* $T_R(\lambda)$,

$$\left|\Pr\left[\mathcal{D}(1^\lambda, \mathbf{ot}_{1,0}) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, \mathbf{ot}_{1,1}) = 1\right]\right| < \epsilon_R(\lambda),$$

  where $\mathbf{ot}_{1,b} \leftarrow \mathsf{OT}_1(1^\lambda, b)$ *for* $b \in \{0, 1\}$, *and the probability is taken over the coins of* OT_1 *and* $\mathcal{D}$.

- ϵ_S***-Statistical Sender Privacy.*** *There exists a (possibly unbounded-time) extractor such that the following holds. For any sequence* $\{\mathbf{ot}_{1,\lambda}, \ell_{0,\lambda}, \ell_{1,\lambda}\}_\lambda$, *define the distribution ensembles* $\{D_{0,\lambda}\}_\lambda$ *and* $\{D_{1,\lambda}\}_\lambda$, *where* $D_{b,\lambda}$ *is defined as follows:*

  1. *Run* $\mathsf{OT}_{\mathsf{extract}}(\mathbf{ot}_{1,\lambda})$ *to obtain* μ.
  2. *If* $b = 0$, *output* $\mathsf{OT}_2(1^\lambda, \ell_{0,\lambda}, \ell_{1,\lambda}, \mathbf{ot}_{1,\lambda})$.
  3. *If* $b = 1$, *set* $\ell'_b = \ell_{b,\lambda}$ *and* $\ell'_{1-b} = 0$ *and output* $\mathsf{OT}_2(1^\lambda, \ell'_0, \ell'_1, \mathbf{ot}_{1,\lambda})$.

  The two ensembles $\{D_{0,\lambda}\}_\lambda$ *and* $\{D_{1,\lambda}\}_\lambda$ *have statistical distance at most* ϵ_S.

In the body of our paper, we will use the following syntax, which reduces trivially to the syntax above.

- $\mathsf{OT}_1(1^\lambda, x; r) \to \mathbf{ot}_1 \to \mathbf{ot}_1$: The receiver's OT_1 algorithm takes a string of choice bits x and produces the receiver's OT message.

- $\mathsf{OT}_2(1^\lambda, \mathsf{lab}, \mathbf{ot}_1; r')$: The sender's OT_2 algorithm takes a list $\mathsf{lab} = \{\mathsf{lab}_{i,b}\}_{i \in [|x|], b \in \{0,1\}}$ of pairs of strings to choose from of length $|x|$ along with the receiver's OT message and produces the sender's OT message.

- $\mathsf{OT}_3(\mathbf{ot}_2; r)$: The receiver's OT_3 takes the sender's OT message and outputs $\{\mathsf{lab}_{i,x_i}\}_{i \in [|x|]}$.

2.3 Two-Round Non-Malleable Commitments

A two-round simultaneous-message non-malleable commitment scheme consists of a tuple of PPT algorithms $(\mathsf{NMC}_1^{\mathsf{send}}, \mathsf{NMC}_1^{\mathsf{recv}}, \mathsf{NMC}_2^{\mathsf{send}})$, which work as follows.

- $\mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, v; r^{\mathsf{send}}) \to c$ takes as input the security parameter 1^λ and the message v and produces the sender's first message committing to val.

- $\mathsf{NMC}_1^{\mathsf{recv}}(1^\lambda; r^{\mathsf{recv}}) \to m$ takes as input the security parameter 1^λ and produces the receiver's (simultaneous) first round message m.

- $\mathsf{NMC}_2^{\mathsf{send}}(1^\lambda, v, m, r^{\mathsf{send}}) \to c_2$ takes as input the security parameter 1^λ, the message v, the receiver's first-round message m, and the randomness r^{send} used to generate the first round message c, and produces the sender's second message c_2.
- $\mathsf{NMC}_{\mathsf{verify}}(1^\lambda, \tau, r^{\mathsf{recv}}) \to 0$ or 1 takes as input a transcript $\tau = (c, m, c_2)$ along with the receiver's randomness r^{recv} and outputs 1 if and only if the transcript is a valid NMC transcript.

In addition, we assume there exists an (unbounded-time) algorithm $\mathsf{NMC}_{\mathsf{extract}}(\tau)$ which takes as input a transcript of the commitment scheme and outputs a value v or τ, such that no opening exists for any value other than v. Thus $\mathsf{NMC}_{\mathsf{extract}}$ defines the unique value committed to by τ.

The definition in this section is taken with small modifications from [KS17a].

We follow the definition of non-malleable commitments introduced by Pass and Rosen [PR05] and further refined by Lin et al. [LPV08] and Goyal [Goy11] (which in turn build on the original definition of [DDN91]). In the real interaction, there is a man-in-the-middle adversary MIM interacting with a committer $\mathcal{C}$ (where $\mathcal{C}$ commits to value v) in the left session, and interacting with receiver $\mathcal{R}$ in the right session. Prior to the interaction, the value $v \in \{0,1\}^n$ is given to C as local input. MIM receives an auxiliary input z, which might contain a-priori information about v. Then the commit phase is executed. Let $\mathsf{MIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}(1^\lambda, v, z)$ denote a random variable that describes the value $\tilde{v}$ committed by the MIM in the right session, jointly with the view of the MIM in the full experiment. In the simulated experiment, a PPT simulator Sim directly interacts with the MIM. Let $\mathsf{Sim}_{\langle C, R \rangle}(1^\lambda, n, z)$ denote the random variable describing the value $\tilde{v}$ committed to by Sim and the output view of Sim. If the tags in the left and right interaction are equal, the value $\tilde{v}$ committed in the right interaction is defined to be $\perp$ in both experiments.

Concurrent non-malleable commitment schemes consider a setting where the MIM interacts with committers in polynomially many (a-priori unbounded) left sessions, and interacts with receiver(s) in up to $\ell(n)$ right sessons. If any of the tags used by MIM (in any right session) are equal to any of the tags in any left session, we set the value committed by the MIM to be $\perp$ for that session. Then we let $\mathsf{MIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}(1^\lambda, v, z)^{\mathsf{many}}$ denote the joint distribution of all the values committed to by the MIM in all right sessions, together with the view of the MIM in the full experiment, and $\mathsf{Sim}_{\langle \mathcal{C}, \mathcal{R} \rangle}(1^\lambda, n, z)^{\mathsf{many}}$ denote the joint distribution of all values committed to by the simulator Sim (with access to the MIM) in all right sessions together with the simulated view of MIM.

Definition 3 ((T, ϵ)-Non-Malleable Commitments w.r.t. Commitment.)**.** *A commitment scheme $\langle \mathcal{C}, \mathcal{R} \rangle$ is said to be (T, ϵ)-non-malleable if for every probabilistic time-T MIM there exists a time-$\mathsf{poly}(T)$ simulator Sim such that for every ensemble $\{(v_\lambda, z_\lambda)\}_\lambda$ of polynomial-length strings v_λ and z_λ, the following ensembles are ϵ-indistinguishable by any time-$T(\lambda)$ adversary:*

$$\{\mathsf{MIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}(v_\lambda, z_\lambda)\}_\lambda \text{ and } \{\mathsf{Sim}_{\langle C, R \rangle}(1^\lambda, |v_\lambda|, z_\lambda)\}_\lambda$$

Definition 4 ((ℓ, T, ϵ)-Concurrent Non-Malleable Commitments w.r.t. Commitment.)**.** *A commitment scheme $\langle \mathcal{C}, \mathcal{R} \rangle$ is said to be (ℓ, T, ϵ)-concurrent non-malleable if for every* *probabilistic time-T* MIM *there exists a time-*$\mathsf{poly}(T)$ *simulator* Sim *such that for every ensemble* $\{(v_\lambda, z_\lambda)\}_\lambda$ *of polynomial-length strings* v_λ *and* z_λ*, the following ensembles are* ϵ*-indistinguishable by any time-*$T(\lambda)$ *adversary:*

$$\{\mathsf{MIM}_{\langle \mathcal{C},\mathcal{R}\rangle}(v_\lambda, z_\lambda)^{\mathsf{many}}\}_\lambda \text{ and } \{\mathsf{Sim}_{\langle C,R\rangle}(1^\lambda, |v_\lambda|, z_\lambda)^{\mathsf{many}}\}_\lambda$$

We say that a commitment scheme is fully concurrent with respect to commitment if it is concurrent for any a-priori unbounded polynomial $\ell(n)$.

In our construction, we will require that the particular two-round simultaneous-message non-malleable commitment we use is binding with respect to the first round. That is, it is impossible to open any transcript starting with $\mathsf{NMC}_1^{\mathsf{send}}(1^n, v)$ to any value other than v. This is very easy to achieve; in particular, we can modify the non-malleable commitment scheme so that the committer additionally sends a non-interactive perfectly-binding commitment to v in the first round, and includes the opening to this commitment as part of its message which it commits to in the non-malleable commitment.

2.4 Garbled Circuits

A garbled circuit scheme consists of a tuple of PPT algorithms $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim}_{\mathsf{Garble}})$, which work as follows.

- $\mathsf{Garble}(1^\lambda, C; r) \to (\tilde{C}, \mathsf{lab})$ takes as input the security parameter 1^λ and a circuit C with λ input bits, and produces a garbled version $\tilde{C}$ along with the list $\mathsf{lab} = \{i, b\}_{i \in [\lambda], b \in \{0,1\}}$ of two labels per input position, corresponding to each bit.

  We denote by lab_x the list $\{i, x_i\}$ corresponding to some input string $x \in \{0,1\}^\lambda$.

- $\mathsf{Eval}(\tilde{C}, \mathsf{lab}_x)$ takes a garbled circuit $\tilde{C}$ and a list $\mathsf{lab}_x = \{i, x_i\}$ of labels corresponding to input x, and outputs the evaluation of $\tilde{C}$ on the input corresponding to the labels.

Definition 5. *A tuple* $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim}_{\mathsf{Garble}})$ *is a* (T, ϵ)*-garbled circuit scheme if the following properties hold:*

- ***Correctness.*** *For all* λ*,* C*, and* $x \in \{0,1\}^\lambda$*,*

  $$\Pr\left[\mathsf{Eval}(\tilde{C}, \mathsf{lab}_x) = C(x) \middle| (\tilde{C}, \mathsf{lab}) \leftarrow \mathsf{Garble}(1^\lambda, C)\right] = 1.$$

  (Recall that lab_x *is defined to be* $\{\mathsf{lab}_{i,x_i}\}_{i \in [\lambda]}$*.)*

- (T, ϵ)***-Privacy.*** *There exists a PPT algorithm* $\mathsf{Sim}_{\mathsf{Garble}}$ *which takes as input* $(1^\lambda, |C|, y)$*, where* 1^λ *is the security parameter and* $y = C(x)$ *is some output of* C*, such that the following holds. For all ensembles* $\{C_\lambda, x_\lambda\}_\lambda$ *and time-*$T(\lambda)$ *distinguishers* $\mathcal{D}$*,*

  $$\left|\Pr\left[\mathcal{D}(1^\lambda, \tilde{C}_\lambda, \mathsf{lab}_{x_\lambda}) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, \hat{C}_\lambda, \hat{\mathsf{lab}}_{x_\lambda}) = 1\right]\right| < \epsilon(\lambda),$$

  where $(\tilde{C}_\lambda, \mathsf{lab}) \leftarrow \mathsf{Garble}(1^\lambda, C_\lambda)$ *and* $(\hat{C}_\lambda, \hat{\mathsf{lab}}) \leftarrow \mathsf{Sim}_{\mathsf{Garble}}(1^\lambda, |C_\lambda|, C_\lambda(x_\lambda))$*.*

2.5 Secure Multiparty Computation

We now define MPC secure against malicious adversaries as well as semi-malicious adversaries, following the general framework given in [?] for defining secure protocols.

A multi-party protocol with n parties is defined with respect to some randomized process, which we call a functionality. This functionality maps n-tuples of inputs to n-tuples of outputs, one for each party. The security of a protocol is defined with respect to a functionality f. In particular, a multiparty protocol for computing a non-reactive functionality f is a protocol running in time polynomial in the total input size and satisfying the following (perfect) correctness requirement: if parties $P_1, \ldots, P_n$ with inputs $x_1, \ldots, x_n$ respectively all run an honest execution of the protocol, then the joint distribution of outputs $y_1, \ldots, y_n$ of the parties is identical to $f(x_1, \ldots, x_n)$.

Communication model. Our protocols will assume all parties have access to a broadcast channel, which any party can use to transmit a message to all other parties.

Defining security. We assume that readers are familiar with standard simulation-based definitions of secure multi-party computation. We provide a self-contained definition for completeness and refer to [?] for a more complete description. The security of a protocol (with respect to a functionality f) is defined by comparing the real-world execution of the protocol with an ideal-world evaluation of f by a trusted party. More concretely, it is required that for every adversary $\mathcal{A}$ which attacks the real execution of the protocol, there exists an adversary Sim, also referred to as a simulator, which can *achieve the same effect* in the ideal world.

The real-world execution. In the real-world execution, the protocol Π is carried out among the n parties, where some subset $\mathcal{C}$ of corrupted parties is controlled by the adversary $\mathcal{A}$. Denote with $\mathsf{real}^{\Pi}_{\mathcal{A}}(\mathcal{C}, (x_1, \ldots, x_n), z)$ a random variable whose value is the output of the execution which is described as follows. $\mathcal{A}$ is initialized with the set $\mathcal{C}$ of corrupted parties along with their inputs $\{x_i\}_{i \in \mathcal{C}}$, and an auxiliary input z. The honest parties are then initialized with the inputs $\{x_i\}_{i \in [n] \setminus \mathcal{C}}$, and then $\mathcal{A}$ performs an execution of Π with the honest parties, providing all messages on behalf of the corrupted parties. At the end of the protocol execution, $\mathcal{A}$ may output an arbitrary function of its view. The output of $\mathsf{real}^{\Pi}_{\mathcal{A}}(\mathcal{C}, (x_1, \ldots, x_n), z)$ is defined to be a tuple consisting of the output of $\mathcal{A}$ along with the outputs of all honest parties.

The ideal-world execution with abort. The ideal-world execution is given with respect to the function f which is computed by an honest execution of Π. In the ideal world, an adversary Sim, called the simulator, interacts with an ideal functionality $\mathcal{F}^f$. Denote with $\mathsf{ideal}^{\mathcal{F}^f}_{\mathsf{Sim}}(\mathcal{C}, (x_1, \ldots, x_n), z)$ the output of the execution which is defined as follows:

- **Initializing the Simulator and Honest Parties:** Sim is initialized with $\mathcal{C}$ and $\{x_i\}_{i \in \mathcal{C}}$. The honest parties P_i, $i \in [n] \setminus \mathcal{C}$, are each initialized with input x_i.

- **Sending inputs to the trusted party:** Every honest party sends its input x_i to $\mathcal{F}^f$, and $\mathcal{F}^f$ records $\tilde{x}_i = x_i$. Sim sends a set $\{\tilde{x}_i\}_{i \in \mathcal{C}}$ of arbitrary inputs, where each $\tilde{x}_i$ is not necessarily equal to x_i.

- **Trusted party sends the corrupted parties' outputs to the adversary:** $\mathcal{F}^f$ now computes $f(\tilde{x}_1, \ldots \tilde{x}_n) \to (y_1, \ldots, y_n)$. It sends $\{y_i\}_{i \in \mathcal{C}}$ to Sim.

- **Adversary chooses which honest parties will abort:** Sim now sends the set $\{\mathsf{instr}_i\}_{i \in [n] \setminus \mathcal{C}}$ to $\mathcal{F}^f$, where for each i, instr_i is either "continue" or "abort". $\mathcal{F}^f$ then sends output y_i to each honest party P_i where instr_i is "continue", and sends output $\perp$ to each honest party P_i where instr_i is "abort".

- **Outputs:** Sim outputs an arbitrary function of its view. The output of the execution is defined to be a tuple consisting of Sim's output along with all outputs of the honest parties.

We now define malicious security for MPC protocols formally in terms of the real-world and ideal-world executions.

Definition 6 (($T_{\mathsf{adv}}, T_{\mathsf{sim}}$)-Malicious-Secure MPC)**.** *We say that an MPC protocol* Π *for a functionality f is* $(T_{\mathsf{adv}}, T_{\mathsf{sim}})$-Malicious secure *if for every nonuniform time-T_{adv} adversary* $(\mathcal{A}, \mathcal{D})$ *there exists a nonuniform time-T_{sim} simulator* Sim *and a negligible function ϵ such that for all inputs $(x_1, \ldots, x_n)$ and every string z,*

$$\left| \Pr\Big[\mathcal{D}(\mathsf{real}^{\Pi}_{\mathcal{A}}(\mathcal{C}, (x_1, \ldots, x_n), z)) = 1\Big] - \Pr\Big[\mathcal{D}(\mathsf{ideal}^{\mathcal{F}^f}_{\mathsf{Sim}}(\mathcal{C}, (x_1, \ldots, x_n), z)) = 1\Big] \right| < \epsilon(k),$$

where $k = |(x_1, \ldots, x_n)|$.

Definition 7 ((polynomial) malicious-secure MPC)**.** *We say that an MPC protocol* Π *is (polynomial) malicious-secure if for all polynomials p there is a polynomial q such that* Π *is (p, q)-malicious secure.*

Definition 8 (SPS malicious-secure MPC)**.** *We say that an MPC protocol* Π *is malicious-secure with super-polynomial simulation (SPS malicious-secure for short) if for all polynomials p there is a sub-exponential function q such that* Π *is (p, q)-malicious secure.*

Concurrent security. The definition of concurrent-secure multiparty computation considers an extension of the above definition where the adversary can spawn and participate simultaneously in many different sessions of the protocol, and can also interact arbitrarily with the environment. We refer the reader to [CLP10, GGJS12] for a detailed definition of concurrent security.

3 The Construction

In this section, we prove the following theorem.

Theorem 2. *Assuming the existence of subexponentially-secure versions of the following primitives:*

- *A two-round SPSS zero knowledge argument system*
- *a two-message concurrent NMC scheme*
- *A two-round statistically-sender private oblivious transfer scheme*
- *A garbled circuit scheme*

there exists a two-round two-party computation protocol for any polynomial-time functionality f, in the plain model with super-polynomial simulation.

We note that each primitive is known from subexponential hardness of LWE. In particular, [BD18] show the existence of two-round statistically-sender-private OT from LWE, and both the SPSS zero-knowledge argument and the NMC scheme of [KS17a] can be instantiated using LWE (see Section 5 for details). Finally garbled circuits can be instantiated using any one-way function, which is known from LWE. Thus we have Theorem 2 as a corollary.

We now describe the construction of two-round two-party computation where both parties receive outputs.

3.1 Required Primitives

First we review the syntax of all the primitives we will use.

Let λ be the security parameter, and we assume 1^λ is an implicit parameter in all the following algorithms.

- A two-round $(T_{\mathsf{sound}}, T_{\mathsf{Sim}}, T_{\mathbf{zk}}, T_L, \epsilon_1, \epsilon_2)$-SPSS ZK argument system

  $$(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_{\mathsf{verify}}, \mathsf{ZK}_{\mathsf{sim}}),$$

  where $T_{\mathsf{sound}}, T_{\mathsf{ZK}_{\mathsf{sim}}}, T_{\mathbf{zk}}, T_L$ are specified below and ϵ_1, ϵ_2 are any negligible functions.

- A two-round $(T_{\mathbf{nmc}}, \epsilon)$-fully-concurrent non-malleable commitment scheme

  $$(\mathsf{NMC}_1^{\mathsf{send}}, \mathsf{NMC}_1^{\mathsf{recv}}, \mathsf{NMC}_2^{\mathsf{send}}),$$

  where $T_{\mathbf{nmc}}$ is specified below and ϵ is any negligible function. In addition, we assume that the extraction algorithm $\mathsf{NMC}_{\mathsf{extract}}$ runs in time $T_{\mathsf{NMC}_{\mathsf{extract}}}$.

- A (T_G, ϵ)-garbled circuit scheme $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim}_{\mathsf{Garble}})$, where T_G is specified below and ϵ is any negligible function.

- A two-round $(T_R, \epsilon_1, \epsilon_2)$-statistically-sender-private OT scheme $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_{\mathsf{extract}})$, where T_R is specified below and ϵ_1, ϵ_2 are any negligible functions. Additionally, we assume the extraction algorithm $\mathsf{OT}_{\mathsf{extract}}$ runs in time $T_{\mathsf{OT}_{\mathsf{extract}}}$.

For the zero-knowledge system, we define a language $L_{i \to j}$ in NP which will be proved by both parties during the 2PC protocol.

$(\mathbf{nmc}_1^{\mathsf{P}_i,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_j,\mathsf{recv}}, \mathbf{nmc}_1^{\mathsf{P}_j,\mathsf{send}}, \mathbf{nmc}_2^{\mathsf{P}_i,\mathsf{send}}, \tilde{C}_i^{\mathsf{P}}, \mathbf{ot}_1^{\mathsf{P}_j}, \mathbf{ot}_2^{\mathsf{P}_i}) \in L_{i \to j}$ **iff:**

There exists $(\mathbf{x}_i, r_c^{\mathsf{P}_i,\mathsf{send}}, r_{gc}^{\mathsf{P}_i}, r_{ot}^{\mathsf{P}_i,\mathsf{send}})$ where

- $\mathbf{nmc}_1^{\mathsf{P}_i,\mathsf{send}} = \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, val; r_c^{\mathsf{P}_i,\mathsf{send}})$ for the value $val = (\mathbf{x}_1, r_{gc}^{\mathsf{P}_i}, r_{ot}^{\mathsf{P}_i,\mathsf{send}})$,
- $\mathbf{nmc}_2^{\mathsf{P}_i,\mathsf{send}} = \mathsf{NMC}_2^{\mathsf{send}}(1^\lambda, val, \mathbf{nmc}_1^{\mathsf{P}_j,\mathsf{recv}}, r_c^{\mathsf{P}_i,\mathsf{send}})$,
- $(\tilde{C}_i^{\mathsf{P}}, \mathsf{lab}) = \mathsf{Garble}(C, r_{gc}^{\mathsf{P}_i})$ for the circuit C defined below, with the hardcoded value set to $(\mathbf{x}_i, \mathbf{nmc}_1^{\mathsf{P}_j,\mathsf{send}})$, and
- $\mathbf{ot}_2^{\mathsf{P}_i} = \mathsf{OT}_2(\mathsf{lab}, \mathbf{ot}_1^{\mathsf{P}_j}, r_{ot}^{\mathsf{P}_i,\mathsf{send}})$, where lab is the family of labels obtained from Garble.

Complexity Hierarchy We require the primitives above satisfy the following complexity hierarchy:

$$\mathsf{poly}(\lambda) \ll T_{\mathsf{sound}} \ll T_{\mathsf{ZK}_{\mathsf{sim}}} \ll T_{\mathbf{nmc}} \ll T_{\mathsf{NMC}_{\mathsf{extract}}} \ll T_R \ll T_{\mathsf{OT}_{\mathsf{extract}}} \ll T_G \ll T_L.$$

Additionally, we require that the language above is decidable in time T_L.

Some Final Notation Let $\mathsf{onlychoices}(x, \mathsf{lab})$ take a string x and a list $\mathsf{lab} = \{\mathsf{lab}_{i,b}\}_{i \in [|x|], b \in \{0,1\}}$ of strings as input and produce the list $\{\mathsf{lab}'_{i,b}\}_{i \in [|x|], b \in \{0,1\}}$, where for each i $\mathsf{lab}'_{i,x_i} = \mathsf{lab}_{i,x_i}$, and $\mathsf{lab}'_{i,1-x_i} = 0$.

3.2 The Protocol

We now describe the protocol for two-round 2PC. Without loss of generality, we describe the actions of Party 1.

2-round 2PC protocol:

In each round, Party 1 performs the following actions.

Round 1:

1. Choose random strings $r_c^{\mathsf{P}_1,\mathsf{send}}, r_c^{\mathsf{P}_1,\mathsf{recv}}, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{recv}}, r_{ot}^{\mathsf{P}_1,\mathsf{send}}$, and $r_{zk}^{\mathsf{P}_1}$ of appropriate sizes.

2. Compute a ZK verifier's message $\mathbf{zk}_1^{\mathsf{P}_1} \leftarrow \mathsf{ZK}_1(1^\lambda; r_{zk}^{\mathsf{P}_1})$.

3. Compute a round-one committer's NMC message

$$\mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{send}} \leftarrow \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, val; r_c^{\mathsf{P}_1,\mathsf{send}}),$$

   where the committed value $val = (\mathbf{x}_1, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}})$ consists of P_1's input along with the randomness which P_1 will use to generate the garbled circuit and OT2 messages in round 2.

4. Compute a round-one receiver's NMC message

$$\mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{recv}} \leftarrow \mathsf{NMC}_1^{\mathsf{recv}}(1^\lambda; r_c^{\mathsf{P}_1,\mathsf{recv}}).$$

5. Compute an OT receiver's message

$$\mathbf{ot}_1^{\mathsf{P}_1} \leftarrow \mathsf{OT}_1(1^\lambda, (\mathbf{x}_1, r_c^{\mathsf{P}_1,\mathsf{send}}, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}}); r_{ot}^{\mathsf{P}_1,\mathsf{recv}}),$$

   where the choice bits $(\mathbf{x}_1, r_c^{\mathsf{P}_1,\mathsf{send}}, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}})$ consist of the randomness $r_c^{\mathsf{P}_1,\mathsf{send}}$ used to generate the round-one sender's NMC message along with the committed values $\mathbf{x}_1, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}}$.

6. Send $(\mathbf{zk}_1^{\mathsf{P}_1}, \mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{recv}}, \mathbf{ot}_1^{\mathsf{P}_1})$ to P_1.

Round 2:

After receiving the first-round message $(\mathbf{zk}_1^{\mathsf{P}_2}$, $\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}}$, $\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{recv}}$, $\mathbf{ot}_1^{\mathsf{P}_2})$ from party 2, party 1 does the following:

1. Compute the sender's second-round NMC message

$$\mathbf{nmc}_2^{\mathsf{P}_1,\mathsf{send}} \leftarrow \mathsf{NMC}_2^{\mathsf{send}}(1^\lambda, val, \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{recv}}, r_c^{\mathsf{P}_1,\mathsf{send}}),$$

   where the committed value $val = (\mathbf{x}_1, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}})$ is as in round 1.

2. Compute the garbled circuit $(\tilde{C}_1^{\mathsf{P}}, \mathsf{lab}) \leftarrow \mathsf{Garble}(1^\lambda, C, r_{gc}^{\mathsf{P}_1})$, where C is the circuit defined below, and the hardcoded values are set to $(\lambda, \mathbf{x}_1, \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}})$.

3. Compute the sender's OT message

$$\mathbf{ot}_2^{\mathsf{P}_1} \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{lab}, \mathsf{OT}_2(1^\lambda, \mathsf{lab}\mathbf{ot}_1^{\mathsf{P}_2}; r_{ot}^{\mathsf{P}_1,\mathsf{send}}),$$

   with the labels lab obtained from the garbling algorithm in the previous step.

4. Compute the prover's ZK message $\mathbf{zk}_2^{\mathsf{P}_1} \leftarrow \mathsf{ZK}_2(1^\lambda, \phi, w, \mathbf{zk}_1^{\mathsf{P}_2})$ for the language $L_{1\to 2}$ with the statement

$$\phi = (\mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{recv}}\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}}, \mathbf{nmc}_2^{\mathsf{P}_1,\mathsf{send}}, \tilde{C}^{\mathsf{P}_1}, \mathbf{ot}_1^{\mathsf{P}_2}, \mathbf{ot}_2^{\mathsf{P}_1})$$

and witness $w = (\mathbf{x}_1, r_c^{\mathsf{P}_1,\mathsf{send}}, r_{gc}^{\mathsf{P}_1}, r_{ot}^{\mathsf{P}_1,\mathsf{send}})$.

5. Send $(\mathbf{nmc}_2^{\mathsf{P}_1,\mathsf{send}}, \tilde{C}_1^{\mathsf{P}}, \mathbf{ot}_2^{\mathsf{P}_1}, \mathbf{zk}_2^{\mathsf{P}_1})$ to P_2.

Output Computation:

After receiving party 2's second-round message $(\mathbf{nmc}_2^{\mathsf{P}_2,\mathsf{send}}, \tilde{C}_2^{\mathsf{P}}, \mathbf{ot}_2^{\mathsf{P}_2}, \mathbf{zk}_2^{\mathsf{P}_2})$, party 1 does the following to compute its output:

1. If the NMC verification algorithm $\mathsf{NMC}_{\mathsf{verify}}(1^\lambda, \tau, r_c^{\mathsf{P}_1,\mathsf{recv}})$ fails with respect to P_2's commitment transcript $\tau = (\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{recv}}, \mathbf{nmc}_2^{\mathsf{P}_1,\mathsf{send}})$, then abort and output $\perp$.

2. Let

$$\phi' = (\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{recv}}\mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{send}}, \mathbf{nmc}_2^{\mathsf{P}_2,\mathsf{send}}, \tilde{C}^{\mathsf{P}_2}, \mathbf{ot}_1^{\mathsf{P}_1}, \mathbf{ot}_2^{\mathsf{P}_2})$$

be the statement which party 2 proves via $\mathbf{zk}_2^{\mathsf{P}_2}$, with respect to language $L_{2\to 1}$. If $\mathsf{ZK}_{\mathsf{verify}}(\phi', \mathbf{zk}_2^{\mathsf{P}_2}, r_{zk}^{\mathsf{P}_1}) = 0$ then abort and output $\perp$.

3. Compute the output labels $\mathsf{lab}' \leftarrow \mathsf{OT}_3(\mathbf{ot}_2^{\mathsf{P}_2}, r_{ot}^{\mathsf{P}_1,\mathsf{recv}})$ of the OT protocol.

4. Output the evaluation $\mathsf{Eval}(\tilde{C}_2^{\mathsf{P}}, \mathsf{lab}')$ of the garbled circuit $\tilde{C}_2^{\mathsf{P}}$ sent by P_2, using the labels lab' obtained in the previous step.

The circuit C which is garbled by Party 1 is as follows.

Circuit C:

Input: $(\mathbf{x}_2, r_c^{\mathsf{P}_2,\mathsf{send}}, r_{gc}^{\mathsf{P}_2}, r_{ot}^{\mathsf{P}_2,\mathsf{send}})$
Hardcoded: $(\lambda, \mathbf{x}_1, \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}})$

```
1. If $\mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}} = \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, (\mathbf{x}_2, r_{gc}^{\mathsf{P}_2}, r_{ot}^{\mathsf{P}_2,\mathsf{send}}); r_c^{\mathsf{P}_2,\mathsf{send}})$, then:
    (a) Return $f(\mathbf{x}_1, \mathbf{x}_2)$
2. Else:
```

(a) Return $\perp$.

4 Security

We now prove Theorem 2 by showing that the protocol above satisfies the definition of concurrent MPC security given in Section 2.5.

Let there be n parties, with a subset of corrupted parties $\mathcal{C} \in [n]$. Consider a PPT adversary $\mathcal{A}$ which spawns a polynomial number of sessions of the protocol described above, where for each session at most one party is corrupt, and schedules messages across the different sessions in an arbitrary order, controlling the inputs and messages of the corrupted parties. The adversary additionally interacts with an environment $\mathcal{Z}$ in an arbitrary manner during the experiment. At the end of the experiment, $\mathcal{Z}$ receives the outputs of all parties in all sessions. We show the existence of an ideal-world adversary (called the "simulator") which produces an interaction with $\mathcal{Z}$ that is indistinguishable from the real-world interaction of $\mathcal{A}$ with $\mathcal{Z}$.

We describe the behavior of the simulator below. In the following, we denote a session by (s, i, j), where s is the session number, and parties P_i and P_j run the 2PC protocol during this session. Without loss of generality we assume P_i is honest and P_j is corrupt, and that $\mathcal{A}$ always asks for the message of P_i in both rounds before sending the message of P_j for that round.

The Concurrent-Secure Simulator:

At the beginning of the experiment, the simulator invokes $\mathcal{A}$. The simulator also initializes a database where it will store, for each session (s, i, j), the messages and extracted values of P_j, the simulator's private state for this session, along with the ideal functionality output for the session. The simulator then responds to $\mathcal{A}$ and $\mathcal{Z}$ in the following manner.

Whenever $\mathcal{A}$ initializes session (s, i, j), do the following to simulate P_i's message to P_j:

1. Choose random strings $r_c^{P_i,\mathsf{send}}, r_c^{P_i,\mathsf{recv}}, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{recv}}, r_{ot}^{P_i,\mathsf{send}}$, and $r_{zk}^{P_i}$ of appropriate sizes. Store all strings as the simulator's private state for session (s, i, j).
2. Compute a ZK verifier's message $\mathbf{zk}_1^{P_i} \leftarrow \mathsf{ZK}_1(1^\lambda; r_{zk}^{P_i})$.
3. Compute a round-one committer's NMC message
$$\mathbf{nmc}_1^{P_i,\mathsf{send}} \leftarrow \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, val; r_c^{P_i,\mathsf{send}})$$
for the value $val = (0, 0, 0)$.

4. Compute a round-one receiver's NMC message

$$\mathbf{nmc}_1^{P_i,\mathsf{recv}} \leftarrow \mathsf{NMC}_1^{\mathsf{recv}}(1^\lambda; r_c^{P_i,\mathsf{recv}}).$$

5. Compute an OT receiver's message $\mathbf{ot}_1^{P_i} \leftarrow \mathsf{OT}_1(1^\lambda, (0,0,0,0); r_{ot}^{P_i,\mathsf{recv}})$, where the choice bits are $(0,0,0,0)$.

6. Send $(\mathbf{zk}_1^{P_i}, \mathbf{nmc}_1^{P_i,\mathsf{send}}, \mathbf{nmc}_1^{P_i,\mathsf{recv}}, \mathbf{ot}_1^{P_i})$ to P_j on behalf of P_i.

Whenever $\mathcal{A}$ sends a first-round message m on behalf of P_j in session (s,i,j), do the following:

1. Parse m as $(\mathbf{zk}_1^{P_j}$, $\mathbf{nmc}_1^{P_j,\mathsf{send}}$, $\mathbf{nmc}_1^{P_j,\mathsf{recv}}$, $\mathbf{ot}_1^{P_j})$. Store m as P_j's first-round message in session (s,i,j).

2. Compute the extracted values $(\mathbf{x}_j, r_c^{P_j,\mathsf{send}}, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}) \leftarrow \mathsf{OT}_{\mathsf{extract}}(\mathbf{ot}_1^{P_j})$ from P_j's OT receiver's message, and save $(\mathbf{x}_j, r_c^{P_j,\mathsf{send}}, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}})$ as P_j's OT receiver value in session (s,i,j).

3. If $\mathbf{nmc}_1^{P_j,\mathsf{send}} = \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, (\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}); r_c^{P_j,\mathsf{send}})$, then send $\mathbf{x}_j$ to the ideal functionality and receive back the evaluation $y = f(\mathbf{x}_i, \mathbf{x}_j)$. If $\mathbf{nmc}_1^{P_j,\mathsf{send}} \neq \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, (\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}); r_c^{P_j,\mathsf{send}})$, set $y = \perp$.

4. Store y as the ideal-world output for P_j in session (s,i,j).

Whenever $\mathcal{A}$ requests a second-round message from honest party P_i in session (s,i,j), do the following:

1. Retrieve P_j's first-round message $m =(\mathbf{zk}_1^{P_j}$, $\mathbf{nmc}_1^{P_j,\mathsf{send}}$, $\mathbf{nmc}_1^{P_j,\mathsf{recv}}$, $\mathbf{ot}_1^{P_j})$ for session (s,i,j).

2. Compute a round-two NMC sender's message

$$\mathbf{nmc}_2^{P_i,\mathsf{send}} \leftarrow \mathsf{NMC}_2^{\mathsf{send}}(1^\lambda, val, \mathbf{nmc}_1^{P_j,\mathsf{recv}}, r_c^{P_i,\mathsf{send}})$$

for the value $val = (0,0,0)$.

3. Compute a simulated garbled circuit $(\tilde{C}_i^P, \mathsf{lab}) \leftarrow \mathsf{Sim}_{\mathsf{Garble}}(1^\lambda, |C|, y; r_{gc}^{P_i})$ using the output y saved previously for session (s,i,j).

4. Compute an OT sender's message $\mathbf{ot}_2^{P_i} \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{onlychoices}(c, \mathsf{lab}), \mathbf{ot}_1^{P_j}, r_{ot}^{P_i,\mathsf{send}})$, where $c = (\mathbf{x}_j, r_c^{P_j,\mathsf{send}}, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}})$ is the saved OT receiver's value for round (s,i,j). Recall that $\mathsf{onlychoices}$ sets all non-chosen labels to 0.

5. Compute a simulated prover's ZK message $\mathbf{zk}_2^{P_i} \leftarrow \mathsf{ZK}_{\mathsf{sim}}(1^\lambda, \phi, \mathbf{zk}_1^{PTwo}, r')$ using the statement

$$\phi = (\mathbf{nmc}_1^{\mathsf{P}_1,\mathsf{send}}, \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{recv}} \mathbf{nmc}_1^{\mathsf{P}_2,\mathsf{send}}, \mathbf{nmc}_2^{\mathsf{P}_1,\mathsf{send}}, \tilde{C}^{\mathsf{P}_1}, \mathbf{ot}_1^{\mathsf{P}_2}, \mathbf{ot}_2^{\mathsf{P}_1})$$

and r' is random.

6. Send $(\mathbf{nmc}_2^{P_i,\mathsf{send}}, \tilde{C}_i^P, \mathbf{ot}_2^{P_i}, \mathbf{zk}_2^{P_i})$ to P_j on behalf of P_i.

Whenever $\mathcal{A}$ sends a second-round message m on behalf of P_j for session (s, i, j), do the following:

1. Parse m as $(\mathbf{nmc}_2^{P_j,\mathsf{send}}, \tilde{C}_j^P, \mathbf{ot}_2^{P_j}, \mathbf{zk}_2^{P_j})$.

2. If the NMC verification algorithm $\mathsf{NMC}_{\mathsf{verify}}(1^\lambda, \tau, r_c^{P_i,\mathsf{recv}})$ fails with respect to P_j's commitment transcript $\tau = (\mathbf{nmc}_1^{P_j,\mathsf{send}}, \mathbf{nmc}_1^{P_i,\mathsf{recv}}, \mathbf{nmc}_2^{P_i,\mathsf{send}})$, then instruct the ideal functionality to deliver $\perp$ to P_i.

3. If $\mathsf{ZK}_{\mathsf{verify}}(s', \mathbf{zk}_2^{P_i}, r_{zk}^{P_j}) = 0$ then instruct the ideal functionality to deliver $\perp$ to P_i.

4. Extract the committed values

$$(\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}) \leftarrow \mathsf{NMC}_{\mathsf{extract}}(\mathbf{nmc}_1^{P_j,\mathsf{send}}, \mathbf{nmc}_1^{P_i,\mathsf{recv}}, \mathbf{nmc}_2^{P_j,\mathsf{send}})$$

from P_j's NMC transcript. If we haven't already queried the ideal functionality, send $\mathbf{x}_j$ to the ideal functionality. Note that because the NMC is perfectly binding after round 1, the value $\mathbf{x}_j$ is identical to the value extracted by $\mathsf{OT}_{\mathsf{extract}}$ during round 2 as long as the identity checked in C holds.

5. Use the values obtained in the previous step to check if the conditions in statement ϕ hold with respect to language $L_{j \to i}$. If they do not hold, output "special abort".

6. If we have not yet aborted, instruct the ideal functionality to deliver the output to P_i.

Whenever $\mathcal{A}$ produces a message to send to the environment, forward the message to $\mathcal{Z}$.
Whenever $\mathcal{Z}$ sends a message, forward to $\mathcal{A}$.

We show the environment $\mathcal{Z}$'s view in the real world is indistinguishable from its view in the ideal world via a series of hybrid games, where the first hybrid $\mathcal{H}_0$ corresponds to the real world and the last hybrid $\mathcal{H}_6$ corresponds to the ideal world. The hybrids are as follows.

Hybrid $\mathcal{H}_0$: In this hybrid, the simulator plays the role of all honest parties in all sessions, and behaves identically to the real-world executions of the protocol. It also forwards all messages between $\mathcal{A}$ and $\mathcal{Z}$.

Hybrid $\mathcal{H}_1$: Here the simulator acts in the same way as in $\mathcal{H}_0$ except that for each honest party P_i's round 2 message during session (s, i, j) it simulates the ZK proof it sends to $\mathcal{A}$. This hybrid now runs in time $\mathsf{poly}(T_{\mathsf{ZK}_{\mathsf{sim}}})$.

Hybrid $\mathcal{H}_2$: The simulator acts in the same way as $\mathcal{H}_1$, except that when computing each honest party P_i's round 1 message during session (s, i, j), it sends the chooser's OT message with choice bits $(0, 0, 0, 0)$ instead of $(\mathbf{x}_i, r_c^{P_i,\mathsf{send}}, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$. This hybrid still runs in time $\mathsf{poly}(T_{\mathsf{ZK}_{\mathsf{sim}}})$.

Hybrid $\mathcal{H}_3$: The simulator acts in the same way as $\mathcal{H}_2$, except for each session (s, i, j), during rounds 1 and 2 it commits to $(0, 0, 0)$ on behalf of P_i instead of $(\mathbf{x}_i, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$. This hybrid still runs in time $\mathsf{poly}(T_{\mathsf{ZK}_{\mathsf{sim}}})$.

Hybrid $\mathcal{H}_4$: The simulator acts in the same way as $\mathcal{H}_3$, except that after receiving P_j's round 2 message during session (s, i, j) it breaks $\mathbf{nmc}_1^{P_j,\mathsf{send}}$ to obtain $(\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}})$ and during the output computation phase outputs "special abort" if the conditions in statement s don't hold. This hybrid now runs in time $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}})$.

Hybrid $\mathcal{H}_5$: The simulator acts in the same way as $\mathcal{H}_4$, except that after receiving P_j's round 1 message during session (s, i, j), it runs $\mathsf{OT}_{\mathsf{extract}}$ on P_j's OT receiver message to obtain the values $(\mathbf{x}_j, r_c^{P_j,\mathsf{send}}, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}})$. If $\mathbf{nmc}_1^{P_j,\mathsf{send}} = \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, (\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}); r_c^{P_j,\mathsf{send}})$, the simulator sends $\mathbf{x}_j$ to the ideal functionality to obtain $f(\mathbf{x}_i, \mathbf{x}_j)$. If $\mathbf{nmc}_1^{P_j,\mathsf{send}} \neq \mathsf{NMC}_1^{\mathsf{send}}(1^\lambda, (\mathbf{x}_j, r_{gc}^{P_j}, r_{ot}^{P_j,\mathsf{send}}); r_c^{P_j,\mathsf{send}})$, the simulator sends the value $\mathbf{x}_j$ extracted using $\mathsf{NMC}_{\mathsf{extract}}$ after receiving P_j's round 2 message to the ideal functionality. It tells the ideal functionality to deliver the output to P1 at the end of session (s, i, j) as long as the session did not abort. This hybrid now runs in time $\mathsf{poly}(T_{\mathsf{OT}_{\mathsf{extract}}})$.

Hybrid $\mathcal{H}_6$: The simulator acts in the same way as $\mathcal{H}_4$, except that for every honest party P_i's second-round message during session (s, i, j), it simulates the generation of the garbled circuit using the saved value y received from the ideal functionality instead of generating it honestly. This final hybrid runs in time $\mathsf{poly}(T_{\mathsf{OT}_{\mathsf{extract}}})$, which is the running time of the ideal-world simulator.

We want to use these hybrids to show the view of $\mathcal{Z}$ is indistinguishable between the real and ideal worlds. There is a problem, though: in $\mathcal{H}_2$ and $\mathcal{H}_3$, the honest parties have no way to obtain its output. This is because the simulator switches the honest parties' OT1 messages to 0 in $\mathcal{H}_2$, which means the real-world method of running the garbled circuit to obtain the output will not work, and the simulator is not yet powerful enough to break the commitment.

Despite this, it is still possible to use this ordering of hybrids to prove indistinguishability. Consider the pair $(s, \mathbf{x}_j, b_i)$, where $\mathbf{x}_j$ is the input committed to by corrupt party P_j during session (s, i, j), and b_i is a bit which denotes whether or not honest party P_i accepts P_j's NMC and zero knowledge proof during the same session. Assuming $\mathcal{A}$ cannot generate a proof for a false statement, this pair determines the output of P_i in session (s, i, j) regardless of whether we are in the real or the ideal world. So to make the proof work, during certain steps we will argue indistinguishability of the tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ between hybrids, where v is the view of $\mathcal{A}$.

The proof is organized as follows.

We first argue computational indistinguishability of the view of $\mathcal{Z}$ between each successive pair of hybrids. Afterwards we argue indistinguishablity of the combined view of $\mathcal{Z}$ along with the output of P1. Before starting, define a "bad" event $\mathcal{E}$ which will be useful in our proofs.

Definition 9. *We define event $\mathcal{E}$ to occur if there exists a session (s, i, j) where both of the following happen:*

1. *P_i accepts P_j's ZK proof*
2. *one of the conditions of the statement s' do not hold w.r.t. $L_{j \to i}$.*

Claim 1. *$\mathcal{E}$ occurs with negligible probability in $\mathcal{H}0$.*

Proof. This follows from the adaptive soundness of the SPSS ZK argument system for languages decidable in time T_L and the fact that $L_{j \to i}$ is decidable in time T_L.

□

Claim 2. *Assuming the zero-knowledge property of the SPSS ZK argument system, the view of $\mathcal{Z}$ between $\mathcal{H}_0$ and $\mathcal{H}_1$ are computationally indistinguishable.*

Proof. We prove the claim via a sequence of subhybrids for each session (s, i, j), where in each subhybrid we switch to a simulated ZK2 message for P_i.

Assume there is a PPT adversary $\mathcal{A}$ who can interact with the simulator and then, given P1's output, distinguish between real and simulated for session (s, i, j). Then we construct a PPT adversary $\mathcal{A}'$ which contradicts the zero-knowledge property of the ZK system.

Fix the randomness used by the adversary, and by the simulator to generate all honest parties' messages before P_i's second-round message. There must be at least one way to fix this randomness such that the advantage of $\mathcal{A}$ is still nonnegligible. This also fixes the statement s (and the witness w for s) which P_i should prove in round 2.

Now we construct $\mathcal{A}'$ to run the experiment with this fixed randomness, and to forward $\mathbf{zk}_1^{P_j}$ to the ZK challenger. Then $\mathcal{A}'$ receives $\mathbf{zk}_2^{P_i}$ which is either a valid proof of s or a simulated one. $\mathcal{A}'$ uses $\mathbf{zk}_2^{P_i}$ as the proof to send to $\mathcal{A}$ instead of generating one itself when generating the second-round message for P_i. It then outputs whatever $\mathcal{A}$ outputs.

$\mathcal{A}$ distinguishes the real and simulated for (s, i, j) even with the round 1 randomness fixed, and this is identical to the experiment described above with the new $\mathcal{A}'$. So $\mathcal{A}'$ is a distinguisher for the zero-knowledge property of the ZK system.

□

Claim 3. *Assuming the zero-knowledge property of the SPSS ZK argument system, $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_1$.*

Proof. Assume there is an adversary $\mathcal{A}$ which causes $\mathcal{E}$ to happen with nonnegligible probability in $\mathcal{H}_1$. Note that by Claim 1 $\mathcal{A}$ cannot cause $\mathcal{E}$ to happen with nonnegligible probability in $\mathcal{H}_0$. We can extract the committed value in time $T_{\mathsf{NMC}_{\mathsf{extract}}}$ for each session to check whether or not $\mathcal{E}$ holds, thus creating a $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}})$-time distinguisher for $\mathcal{H}_0$ and $\mathcal{H}_1$, contradicting Claim 2, since $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}}) \ll T_{\mathbf{zk}}$.

□

Claim 4. *Assuming the chooser's security of the OT scheme, the tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ between $\mathcal{H}_1$ and $\mathcal{H}_2$ is computationally indistinguishable.*

Proof. We prove the claim via a sequence of subhybrids for each session (s, i, j), where in each subhybrid we switch P_i's OT1 message to 0.

Assume there is a PPT adversary $\mathcal{A}$ who can interact with the simulator and then, given $\{(s, \mathbf{x}_j, b_i)\}_s$ in addition to its view v at the end of the interaction, distinguishes between the subhybrid for some (s, i, j) and the previous subhybrid with nonnegligible probability. We use $\mathcal{A}$ to build an adversary $\mathcal{A}'$ for the OT chooser's security game. For simplicity of exposition, we first assume that $\mathcal{A}$ distinguishes only given its view v. Once we have established the reduction in this case, we extend it to the case where $\mathcal{A}$ also receives $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$.

Fix the randomness $(r_c^{P_i,\mathsf{send}}, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$ generated on behalf of P_i for session (s, i, j). There must be at least one such fixed value for which $\mathcal{A}$ still distinguishes with nonnegligible probability. Let $\mathcal{A}'$ run the experiment identically to the previous subhybrid with the randomness above fixed to this particular value, except that instead of computing $\mathbf{ot}_1^{P_i}$ directly it receives this value from the OT challenger. The challenger either computes the OT with choice bits $(r_c^{P_i,\mathsf{send}}, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$ or $(0, 0, 0, 0)$.

Assuming $(r_c^{P_i,\mathsf{send}}, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$ is fixed, this experiment is identical to the previous subhybrid in the first case and the subhybrid for (s, i, j) in the second case. So if $\mathcal{A}$ successfully distinguishes then $\mathcal{A}'$ does as well. This contradicts chooser's security of the OT, since $T_{\mathsf{ZK}_{\mathsf{sim}}} \ll T_R$.

To extend to the case where $\mathcal{A}$ also receives $\{(s, \mathbf{x}_j, b_i)\}_s$, note that we can break the commitments of each of the corrupted parties in time $T_{\mathsf{NMC}_{\mathsf{extract}}}$ to retrieve each corrupted input $\mathbf{x}_j$, and b_i is known already by the experiment. Passing these to the adversary we obtain a $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}})$-time distinguisher, which still contradicts chooser's security of the OT, since $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}}) \ll T_R$.

□

Claim 5. *Assuming $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_1$ and the nonmalleability of the commitment scheme, $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_2$.*

Proof. Assume there is an adversary $\mathcal{A}$ which causes $\mathcal{E}$ to happen with nonnegligible probability in $\mathcal{H}_2$. Note that by Claim 3 $\mathcal{A}$ cannot cause $\mathcal{E}$ to happen with nonnegligible probability in $\mathcal{H}_1$. We can break the corrupted parties' commitments each in time $T_{\mathsf{NMC}_{\mathsf{extract}}}$ to create a $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}})$-time distinguisher for $\mathcal{H}_1$ and $\mathcal{H}_2$, contradicting Claim 4, since $\mathsf{poly}(T_{\mathsf{NMC}_{\mathsf{extract}}}) \ll T_R$.

□

Claim 6. *Assuming non-malleability of the commitment scheme, the tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ between $\mathcal{H}_2$ and $\mathcal{H}_3$ is computationally indistinguishable.*

Proof. We prove the claim via a sequence of subhybrids for each session (s, i, j), where in each subhybrid we switch to a NMC of 0 for P_i.

Assume there is a PPT adversary $\mathcal{A}$ who can interact with the simulator and then, given $\{(s, \mathbf{x}_j, b_i)\}_s$ in addition to its view v at the end of the interaction, distinguishes between the previous hybrid and the hybrid for (s, i, j) with nonnegligible advantage. Then we create a $T_{\mathsf{ZK}_{\mathsf{sim}}}$-time $\mathcal{A}'$ which contradicts the non-malleability property of the commitment scheme. Note that $\{(s, \mathbf{x}_j, b_i)\}_s$ is computable directly from the output of the non-malleability game, since each corrupted party P_j commits to x_j (i.e., these are part of the RHS committed values).

Fix the randomness $r_{gc}^{P_i}$ and $r_{ot}^{P_i,\mathsf{send}}$ generated for P_i in session (s,i,j). There must be some such fixed values where $\mathcal{A}$ still distinguishes between the two subhybrids with nonnegligible advantage. We create $\mathcal{A}'$ as follows. $\mathcal{A}'$ runs the experiment identically to the previous subhybrid, except that the values $r_{gc}^{P_i}$ and $r_{ot}^{P_i,\mathsf{send}}$ are fixed to maximize the probability of distinguishing, and the following changes are made to the nonmalleable commitment interactions. When computing P_i's round 1 message, instead of computing $\mathbf{nmc}_1^{P_i,\mathsf{send}}$, it receives this value from the challenger for the NMC game and forwards it to $\mathcal{A}$. It forwards $\mathbf{nmc}_1^{P_j,\mathsf{recv}}$ which it receives from $\mathcal{A}$ to the challenger as well. When computing P_i's round 2 message, it receives $\mathbf{nmc}_2^{P_i,\mathsf{send}}$ from the challenger and forwards it to $\mathcal{A}$ again. The NMC challenger commits to either $(\mathbf{x}_i, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$ or $(0,0,0)$.

If the challenger commits to $(\mathbf{x}_i, r_{gc}^{P_i}, r_{ot}^{P_i,\mathsf{send}})$ then the experiment is identical to the subhybrid directly preceding the subhybrid for session (s,i,j), and if the challenger commits to $(0,0,0)$ then the experiment is identical to the subhybrid for (s,i,j) (with some fixed randomness, as described above). Thus $\mathcal{A}'$ wins the hiding security game of the commitment scheme with nonnegligible probability, contradicting hiding of the commitment scheme, since $\mathsf{poly}(T_{\mathsf{ZK}_{\mathsf{sim}}}) \ll T_{\mathbf{nmc}}$.

□

Claim 7. *Assuming $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_2$ and the nonmalleability of the commitment scheme, $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_3$.*

Proof. Assume that $\mathcal{E}$ occurs with non-negligible probability in $\mathcal{H}_3$. We can construct an adversary $\mathcal{A}'$ in the same way as in Claim 6, playing the role of the adversary in a full nonmalleability game. By the nonmalleability property of the commitment scheme, the joint view of $\mathcal{A}'$ combined with the values it committed to are indistinguishable regardless of what the challenger commits to. If we have both the view of $\mathcal{A}'$ along with the values committed to it is easy to check if $\mathcal{E}$ occured. If $\mathcal{E}$ occurs with nonnegligible probability in $\mathcal{H}_2$ then by checking $\mathcal{E}$ we have a $\mathsf{poly}(T_{\mathsf{ZK}_{\mathsf{sim}}})$-time distinguisher which contradicts nonmalleability of the NMC.

□

Claim 8. *Assuming $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_3$, the tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ between $\mathcal{H}_3$ and $\mathcal{H}_4$ are computationally indistinguishable.*

Proof. The only difference between $\mathcal{H}_3$ and $\mathcal{H}_4$ is that we break all corrupted parties' commitments and output "special abort" if at any point $\mathcal{E}$ occurred. So the only time the two hybrids are distinguishable is if $\mathcal{E}$ occurs. Thus indistinguishability follows from Claim 7.

□

Note that from this point onward, proving hybrid indistinguishability is sufficient for proving $\mathcal{E}$ occurs with negligible probability, since every hybrid now checks $\mathcal{E}$ explicitly.

Claim 9. *The tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ between $\mathcal{H}_4$ and $\mathcal{H}_5$ are computationally indistinguishable.*

Proof. This follows trivially from the fact that the view of $\mathcal{A}$ is identical between $\mathcal{H}_4$ and $\mathcal{H}_5$. □

Claim 10. *Assuming $\mathcal{E}$ happens with negligible probability in $\mathcal{H}_1$ and $\mathcal{H}_5$, the view of the environment $\mathcal{Z}$ between $\mathcal{H}_1$ and $\mathcal{H}_5$ is computationally indistinguishable.*

Proof. By the previous claims the tuple $(v, \{(s, \mathbf{x}_j, b_i)\}_s)$ is indistinguishable between these hybrids. Assuming $\mathcal{E}$ did not happen, in both hybrids the output of each honest party P_i during session (s, i, j) is $f(\mathbf{x}_i, \mathbf{x}_j)$ if $b = 1$ and $\perp$ if $b = 0$. To see why this is the case when $b = 1$, note that the value $\mathbf{x}_j$ extracted by $\mathsf{OT}_{\mathsf{extract}}$ after round 1 is identical to the value extracted for $\mathbf{x}'_j$ by $\mathsf{NMC}_{\mathsf{extract}}$ after round 2. Assuming $\mathcal{E}$ does not occur, P1 outputs $\mathbf{x}'_j$ in $\mathcal{H}_1$, and P1 outputs $\mathbf{x}_j$ in $\mathcal{H}_5$.

Thus the claim follows from the fact that $\mathcal{E}$ occurs with negligible probability in $\mathcal{H}_5$, which follows from Claim 9. □

Claim 11. *Assuming security of the garbled circuit scheme and statistical sender's security of the OT, the view of $\mathcal{Z}$ between $\mathcal{H}_5$ and $\mathcal{H}_6$ is computationally indistinguishable.*

Proof. We consider a subhybrid $\mathcal{H}'_5$ which acts similarly to $\mathcal{H}_5$ except that when generating the second-round message for each honest P_i during session (s, i, j), it uses $\mathsf{onlychoices}$ to zero out the labels given by the honest party in OT2 which do not correspond to the adversary's input.

This claim then then follows from the next two claims. □

Claim 12. *Assuming statistical sender's security of the OT, the view of the environment $\mathcal{Z}$ between $\mathcal{H}_5$ and $\mathcal{H}'_5$ are statistically indistinguishable.*

Proof. We prove the claim via a sequence of subhybrids for each session (s, i, j), where in each subhybrid we switch the OT2 message of P_i to zero out non-chosen labels.

Assume there is an adversary $\mathcal{A}$ who can interact with the simulator and then cause the environment to distinguish between the subhybrid for some session (s, i, j) and the preceding subhybrid with nonnegligible probability. We use $\mathcal{A}$ and $\mathcal{Z}$ to build an adversary $\mathcal{A}'$ for the OT sender's security game.

Fix the randomness $r_{gc}^{P_i}$ used to generate P_i's garbled circuit which it sends as part of its second-round message. There must be at least one such fixed value for which $\mathcal{Z}$ still distinguishes with nonnegligible probability. Let $\mathcal{A}'$ run the experiment identically to the subhybrid preceding (s, i, j) with $r_{gc}^{P_i}$ fixed to this value, except that it passes the OT1 message $\mathbf{ot}_1^{P_j}$ generated by $\mathcal{A}$ to the OT challenger. The OT challenger then either responds with an $\mathbf{ot}_2^{P_i}$ corresponding to the same labels in $\mathcal{H}_5$, or breaks $\mathbf{ot}_1^{P_j}$ and zeros out the labels which do not correspond to the adversary's input. $\mathcal{A}'$ then outputs the output of $\mathcal{Z}$.

Assuming $r_{gc}^{P_i}$ is fixed, this experiment is identical to the preceding hybrid in the first case and the subhybrid for (s, i, j) in the second case. So if $\mathcal{Z}$ successfully distinguishes then $\mathcal{A}'$ does as well. □

Claim 13. *Assuming security of the garbled circuit scheme, the view of $\mathcal{Z}$ between $\mathcal{H}_5'$ and $\mathcal{H}_6$ is computationally indistinguishable.*

Proof. We prove the claim via a sequence of subhybrids for each session (s, i, j), where in each subhybrid we switch the garbled circuit of P_i to be simulated.

Assume there is an adversary $\mathcal{A}$ who can interact with the simulator and then cause $\mathcal{Z}$ to distinguish between the subhybrid directly preceding the one for some session (s, i, j) and the subhybrid for (s, i, j) with nonnegligible probability. We use $\mathcal{A}$ and $\mathcal{Z}$ to build a $\mathsf{poly}(T_{\mathsf{OT}_{\mathsf{extract}}})$-time adversary $\mathcal{A}'$ that contradicts security of the garbled circuit.

Fix the randomness used by $\mathcal{A}$ and the randomness used by the simulator in generating all rounds preceding P_i's second-round message during session (s, i, j). There must be some such fixed randomness such that $\mathcal{Z}$ still distinguishes with nonnegligible probability. This also fixes the circuit which the honest party P_i garbles along with P_j's OT1 input in session (s, i, j).

Let $\mathcal{A}'$ work in the same way as the subhybrid preceding (s, i, j) except that it receives a garbled circuit and labels $(\tilde{C}_1^{\mathsf{P}}, labels)$ from the challenger, which it uses as the garbled circuit and labels for P_i in session (s, i, j). The challenger either computes the garbled circuit honestly or simulates using the output $f(\mathbf{x}_i, \mathbf{x}_j)$, which is fixed because of the fixed randomness. $\mathcal{A}'$ finally outputs the output of $\mathcal{Z}$.

Based on what the challenger does, the experiment is either identical to the subhybrid preceding (s, i, j) or the (s, i, j) subhybrid (with the adversary's randomness fixed). This means $\mathcal{Z}$ distinguishes with nonnegligible probability and thus so does $\mathcal{A}'$, contradicting security of the garbled circuit, since $\mathsf{poly}(T_{\mathsf{OT}_{\mathsf{extract}}}) \ll T_G$.

□

5 Instantiating the Non-Malleable Commitment of [KS17a] using LWE

In this section, we list all primitives used in each part of the construction of fully concurrent non-malleable commitments of [KS17b], along with how to instantiate each primitive using subexponential hardness of LWE.

Two-round extractable commitments are constructed using the following primitives:

- A non-interactive perfectly binding commitment. *Can be obtained from LWE using one of the LWE-based PKE schemes.*
- Two-round statistically-sender-private oblivious transfer. *Known from LWE [BD18].*
- Yao's garbled circuits. *Symmetric-key encryption is known from LWE.*
- Two-round zero knowledge with super-polynomial simulation. *Known from LWE [BFJ+20].*

Two-round SPS strong zero knowledge arguments are constructed using the following primitives:

- A zap. *Known from LWE [BFJ+20].*

- The two-round extractable commitment. *See above.*
- A trapdoor for the prover to use. *Can use an instance of SIS.*

Two-round constant-tag non-malleable commitments are constructed using the following primitives:

- The two-round extractable commitment. *See above.*
- A non-interactive perfectly binding commitment. *Can be obtained from LWE using one of the LWE-based PKE schemes.*

One-one non-malleable commitments in two rounds are constructed using the following primitives:

- The two-round constant-tag non-malleable commitment. *See above.*
- The two-round SPS strong ZK. *See above.*

One-one simulation-sound zero knowledge in two rounds is constructed using the following primitives:

- The one-one non-malleable commitment. *See above.*
- A zap. *Known from LWE [BFJ+20].*
- A trapdoor for the prover to use. *Can use an instance of SIS.*

Fully-concurrent non-malleable commitments in two rounds are constructed using the following primitives:

- The one-one simulation-sound zero knowledge argument. *See above.*
- The constant-tag non-malleable commitment. *See above.*

References

[ABG+21] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Two-round maliciously secure computation with super-polynomial simulation. In *TCC*, 2021.

[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, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[AGIS14] Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014*, pages 646–658, Scottsdale, AZ, USA, November 3–7, 2014. ACM Press.

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 191–225, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[AJS18] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. Cryptology ePrint Archive, Report 2018/615, 2018. `https://eprint.iacr.org/2018/615`.

[AMR21] Behzad Abdolmaleki, Giulio Malavolta, and Ahmadreza Rahimi. Two-round concurrently secure two-party computation. *IACR Cryptol. ePrint Arch.*, 2021.

[AP20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 110–140, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[AS17] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 152–181, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 406–435, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 79–109, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. `https://eprint.iacr.org/2020/1024`.

[BFJ$^+$20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[BGH$^+$15] Zvika Brakerski, Craig Gentry, Shai Halevi, Tancrède Lepoint, Amit Sahai, and Mehdi Tibouchi. Cryptanalysis of the quadratic zero-testing of GGH. Cryptology ePrint Archive, Report 2015/845, 2015. `https://eprint.iacr.org/2015/845`.

[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, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.

[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, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

[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, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

[BGK$^+$14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 221–238, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[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, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

[BIJ$^+$20] James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. Affine determinant programs: A framework for obfuscation and witness encryption. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 82:1–82:39, Seattle, WA, USA, January 12–14, 2020. LIPIcs.

[BL18] Nir Bitansky and Huijia Lin. One-message zero knowledge and non-malleable commitments. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 209–234, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

[BMSZ16] Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 764–791, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th FOCS*, pages 345–354, Berkeley, CA, USA, October 21–24, 2006. IEEE Computer Society Press.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 1–25, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. Cryptology ePrint Archive, Report 2014/930, 2014. `https://eprint.iacr.org/2014/930`.

[CCG$^+$20] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 291–319, Durham, NC, USA, November 16–19, 2020. Springer, Heidelberg, Germany.

[CGH$^+$15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 247–266, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *51st FOCS*, pages 541–550, Las Vegas, NV, USA, October 23–26, 2010. IEEE Computer Society Press.

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. Cryptology ePrint Archive, Report 2015/934, 2015. `https://eprint.iacr.org/2015/934`.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 267–286, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pages 542–552, New Orleans, LA, USA, May 6–8, 1991. ACM Press.

[DGG$^+$16] Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Pratyay Mukherjee. Obfuscation from low noise multilinear maps. Cryptology ePrint Archive, Report 2016/599, 2016. `https://eprint.iacr.org/2016/599`.

[FJK21] Rex Fernando, Aayush Jain, and Ilan Komargodski. Maliciously-secure MrNISC in the plain model. *IACR Cryptol. ePrint Arch.*, 2021.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC*, 2015.

[GGJS12] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Concurrently secure computation in constant rounds. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 99–116, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[GJK18] Craig Gentry, Charanjit S. Jutla, and Daniel Kane. Obfuscation using tensor products. Cryptology ePrint Archive, Report 2018/756, 2018. `https://eprint.iacr.org/2018/756`.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 448–476, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 695–704, San Jose, CA, USA, June 6–8, 2011. ACM Press.

[GP20] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. Cryptology ePrint Archive, Report 2020/1010, 2020. `https://eprint.iacr.org/2020/1010`.

[GS17] Sanjam Garg and Akshayaram Srinivasan. Garbled protocols and two-round MPC from bilinear maps. In Chris Umans, editor, *58th FOCS*, pages 588–599, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.

[Hal15] Shai Halevi. Graded encoding, variations on a scheme. Cryptology ePrint Archive, Report 2015/866, 2015. https://eprint.iacr.org/2015/866.

[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, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

[HJ15] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. Cryptology ePrint Archive, Report 2015/301, 2015. https://eprint.iacr.org/2015/301.

[HJL21] Sam Hopkins, Aayush Jain, and Huijia Lin. Counterexamples to circular security-based io. In *CRYPTO*, 2021.

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 251–281, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[JLS21a] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. *IACR Cryptol. ePrint Arch.*, 2021.

[JLS21b] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, pages 60–73. ACM, 2021.

[Khu21] Dakshita Khurana. Non-interactive distributional indistinguishability (NIDI) and non-malleable commitments. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 186–215, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[KS17a] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Chris Umans, editor, *58th FOCS*, pages 564–575, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.

[KS17b] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. Cryptology ePrint Archive, Report 2017/291, 2017. https://eprint.iacr.org/2017/291.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 28–57, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Lin17] Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 599–629, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[LM18] Huijia Lin and Christian Matt. Pseudo flawed-smudging generators and their application to indistinguishability obfuscation. Cryptology ePrint Archive, Report 2018/646, 2018. `https://eprint.iacr.org/2018/646`.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 571–588, San Francisco, CA, USA, March 19–21, 2008. Springer, Heidelberg, Germany.

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 630–660, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *57th FOCS*, pages 11–20, New Brunswick, NJ, USA, October 9–11, 2016. IEEE Computer Society Press.

[MF15] Brice Minaud and Pierre-Alain Fouque. Cryptanalysis of the new multilinear map over the integers. Cryptology ePrint Archive, Report 2015/941, 2015. `https://eprint.iacr.org/2015/941`.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[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, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[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, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany.

[PR05] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th FOCS*, pages 563–572, Pittsburgh, PA, USA, October 23–25, 2005. IEEE Computer Society Press.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 500–517, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[WW20] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. Cryptology ePrint Archive, Report 2020/1042, 2020. https://eprint.iacr.org/2020/1042.