Two-Round Adaptively Secure MPC from Isogenies, LPN, or CDH

Navid Alamati[1], Hart Montgomery[2], Sikhar Patranabis[3], Pratik Sarkar[4]

[1] UC Berkeley and Visa Research[***]
[2] Fujitsu Research of America
[3] ETH Zürich and Visa Research[†]
[4] Boston University

Abstract. We present a new framework for building round-optimal (two-round) *adaptively* secure MPC. We show that a relatively weak notion of OT that we call *indistinguishability OT with receiver oblivious sampleability* (r-iOT) is enough to build two-round, adaptively secure MPC against *malicious* adversaries in the CRS model. We then show how to construct r-iOT from CDH, LPN, or isogeny-based assumptions that can be viewed as group actions (such as CSIDH and CSI-FiSh). This yields the first constructions of two-round adaptively secure MPC against malicious adversaries from CDH, LPN, or isogeny-based assumptions. We further extend our non-isogeny results to the plain model, achieving (to our knowledge) the first construction of two-round adaptively secure MPC against semi-honest adversaries in the plain model from LPN.

Our results allow us to build two-round adaptively secure MPC against malicious adversaries from essentially all of the well-studied assumptions in cryptography. In addition, our constructions from isogenies or LPN provide the first post-quantum alternatives to LWE-based constructions for round-optimal adaptively secure MPC. Along the way, we show that r-iOT also implies non-committing encryption (NCE), thereby yielding the first constructions of NCE from isogenies or LPN.

1 Introduction

Secure multiparty computation (MPC) allows mutually distrusting parties to jointly evaluate functions of their secret inputs in a manner that doesn't reveal any information outside of the final output. More precisely, an MPC protocol involves n parties $P_1, \ldots, P_n$ with private inputs $x_1, \ldots, x_n$ such that, at the end of the protocol, each party P_i learns an output of the form $f_i(x_1, \ldots, x_n)$ but nothing else about the private inputs of any other party.

MPC has been extensively studied since the 1980s [Yao86, GMW87] and is currently used in practice for a wide variety of applications, such as privacy-preserving studies for social good [LJA+18], privacy-preserving online advertising [IKN+17], distributed key management [unb], and securely instantiating blockchain protocols [CCD+20].

[***] Most of the work was done while the author was affiliated with UC Berkeley.
[†] Most of the work was done while the author was affiliated with ETH Zürich.

MPC constructions are closely related to (and often based upon) another widely studied primitive called *oblivious transfer* (OT) [Rab05, EGL82]. Informally speaking, an OT protocol involves a *sender* holding two messages m_0 and m_1, and a receiver holding a bit b. At the end of the protocol, the receiver should only learn the message m_b and nothing about m_{1-b}, while the sender should learn nothing about the bit b. Due to its wide range of applications, OT has been studied extensively in a long line of works [NP01, PVW08, BD18, FMV19, DGH+20, LGdSG21, CSW20, ADMP20].

Models and Round Complexity. Given the ubiquity of MPC in cryptography, it is no surprise that MPC protocols have been studied in many different security models. Examples of such models include *semi-honest/malicious* as well as *static/adaptive* adversarial corruptions. MPC has also been studied in a variety of computational models such as the plain model and the common reference string (CRS) model. An important feature of any MPC protocol is its *round complexity* (i.e., the number of rounds of communication between the parties during protocol execution). Minimal round complexity is desirable when communication time dominates computational cost, which is the case in many practical protocols. So, designing *round-optimal* MPC protocols is widely regarded to be an important topic in MPC research.

The Static Corruption Model. In the *static* corruption model for MPC, the adversary is allowed to corrupt a pre-determined set of parties. A long line of works have shown how to design round-optimal MPC protocols in this model from a variety of assumptions in the CRS model [GGHR14, MW16, CPV17a]. Notably, [BL18, GS18] showed how to construct two-round MPC protocols from two-round OT protocols in different security models and computational settings.

In terms of concrete computational assumptions, two-round maliciously secure OT protocols in the static corruption model have been constructed from DDH, QR/DCR, and LWE [NP01, PVW08, HK12, BD18]. More recently, such OT protocols have been designed from the CDH and LPN assumptions [DGH+20], as well as from isogenies of elliptic curves [ADMP20]. To summarize, we can currently build round-optimal maliciously secure MPC in the static corruption model from essentially all of the commonly used computational assumptions.

Limitations of the Static Corruption Model. Unfortunately, the static corruption model for MPC is not strong enough for certain real-world applications. In particular, the static corruption model does not provide security against "hacking attacks" where an adversary might adaptively corrupt parties at different stages of the protocol. For instance, what happens if the adversary seizes control of the parties' machines through backdoor access? Secure erasures of the party's state upon corruption is one possible solution to tackle such an attack. However, it is an impractical solution as argued by [CFGN96] since it requires the party to detect an attack and honestly execute its erasure of internal state. This motivates designing MPC protocols that are secure in the *adaptive* corrup-

tion model without relying on secure erasures. In this work we refer to adaptive security in the non-erasure model as adaptive security.

The Adaptive Corruption Model. In the *adaptive* corruption model for MPC, the adversary is allowed to dynamically corrupt any set of parties at any time during the protocol execution. Canetti *et al.* [CDD+04] presented the first formal investigation of the adaptive corruption model for MPC, and the relationships between adaptive security and static security in several models of computation. Garay *et al.* [GWZ09] showed how to construct adaptively secure two-party computation protocols in a generic manner from OT protocols satisfying a weaker notion of *semi-adaptive security*; they also showed how to obtain semi-adaptively secure OT protocols from somewhat non-committing encryption (NCE), which is a weaker variant of standard NCE [CLOS02]. Subsequently, Hazay *et al.* [HV15] showed that adaptively secure MPC protocols can be obtained from minimal assumptions like trapdoor simulatable public key encryption (PKE).

However, the scenario is different once round optimality is taken into consideration. It is currently open to design round-optimal maliciously secure MPC protocols even from certain commonly used computational assumptions such as CDH, LPN and isogeny-based assumptions.[5] Initial works on two round, adaptively secure MPC relied on indistinguishability obfuscation (and other standard assumptions) [CGP15, GP15, CPV17a, CsW19] or assuming secure erasures[6] [CsW19] of the party's internal states.

The work of Benhamouda *et al.* [BLPV18] was the first to show how to construct round-optimal adaptively universal composability (UC) [Can01] secure MPC protocols from certain standard computational assumptions without obfuscation and erasures. More concretely, they established the following:

- Against *semi-honest* adversaries, adaptively UC-secure two-round MPC in the plain model is implied by non-committing encryption (NCE) [CLOS02], which in turn can be built from CDH/DDH, LWE, and RSA [CDMW09].
- Against *malicious* adversaries, adaptively UC-secure two-round MPC in the CRS model can be built from a certain kind of two-round statically secure OT protocol with additional "oblivious sampleability" properties, which in turn can be based on DDH, QR, and LWE.

The recent work of [CSW20] constructs a two round adaptively secure MPC protocol based on the DDH assumption. It is currently open to construct round-optimal (i.e., two-round) maliciously secure MPC protocols in the adaptive corruption model from commonly studied assumptions such as CDH, LPN and

[5] Note that constant round maliciously secure MPC against adaptive corruptions can only be achieved in the CRS model; see [GS12] for results establishing the impossibility of maliciously secure adaptive MPC in the plain model from black-box simulation.

[6] The secure erasures model allows erasing the internal state of an honest party when its gets adaptively corrupted by the adversary. It is a strictly weaker model than the one we consider, where erasing the party's state is not allowed.

isogeny-based assumptions. In particular, the constructions of Benhamouda *et al.* [BLPV18] crucially rely on certain primitives such as "obliviously sampleable" smooth projective hash functions (SPHFs) and "augmented" non-committing encryption (NCE) that are not known from some or all of these assumptions. More generally, it is not known how to construct such MPC protocols from a *single* generic primitive that can be built from commonly used computational assumptions.

Moreover, there are motivating concerns about efficient quantum computing and adaptive MPC. Currently, the only plausibly post-quantum secure constructions [BLPV18, CsW19] of two-round maliciously secure MPC protocols in the adaptive corruption model are based on LWE. This lack of diversity in post-quantum constructions is potentially concerning since a major advance in lattice cryptanalysis could substantially degrade (or in the worst case, invalidate) the security of LWE-based constructions for all practical parameter sets. Notably, the recent NIST competition to standardize post-quantum cryptosystems [CJL+16, AAAS+19, AASA+20] considers a wider class of post-quantum assumptions, including isogeny-based assumptions. In this paper, we ask the following question:

Can we construct two round adaptively UC-secure MPC protocols from a wider class of assumptions, such as CDH, LPN, and isogeny-based assumptions?

1.1 Our Contributions

We answer this above question in the affirmative. We establish a new route to achieving two round maliciously UC-secure MPC protocols in the adaptive corruption setting that relies on potentially weaker (or "less structured") cryptographic primitives as compared to those used by Benhamouda *et al.* [BLPV18]. We also show how to instantiate these primitives from CDH, LPN, and certain families of isogeny-based assumptions (such as CSIDH [CLM+18] and CSI-FiSh [BKV19]). Our results thus establish the feasibility of realizing adaptively secure MPC from essentially *all* commonly used cryptographic assumptions.

We present our results in the "local" CRS model where every session of protocol execution has a local independently sampled CRS string. This is the same model in which Benhamouda *et al.* [BLPV18] described their constructions and proofs. The only other work [CSW20] in this setting is in the single common random string model, but it is solely based on DDH. We note here that Choi *et el.* [CKWZ13] achieved efficient, adaptively secure, composable OT protocols with a single, global CRS, albeit from a different set of concrete assumptions as compared to what we consider in this paper.

Our Ingredients. Our constructions of two-round, adaptively UC-secure MPC essentially rely on a *single* building block, which we refer to as *indistinguishability OT with receiver oblivious sampleability* (r-iOT). Informally, r-iOT is a two-message OT protocol that satisfies indistinguishability security [DGH+20] against the sender and the receiver in the *static* corruption model, while also

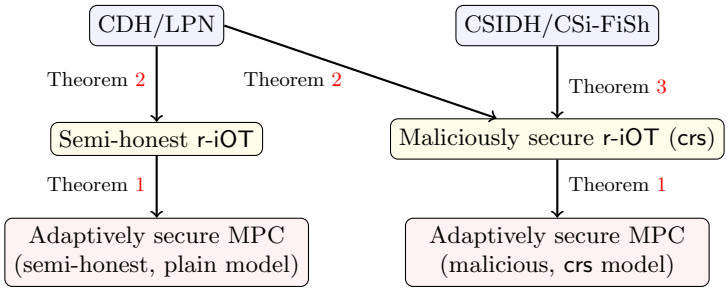

Fig. 1. A simplified overview of our results

satisfying an additional property called "receiver oblivious sampleability". At a high level, this property requires that it is possible to obliviously sample the OT receiver's message (without knowledge of any secret randomness and receiver's choice bit). This property also requires an algorithm for claiming that an honestly generated receiver's message was, in fact, obliviously sampled.

We note that the concept of receiver oblivious sampleable OT was introduced and used in their constructions by Benhamouda *et al.* [BLPV18]. However, our constructions rely on a *strictly weaker* set of properties for our starting r-iOT protocol. First of all, the constructions in [BLPV18] assume that the starting OT protocol satisfies (simulation-based) UC-security against a semi-honest sender and a malicious receiver in the static corruption model. On the other hand, our starting r-iOT protocol is only required to achieve a strictly weaker notion of indistinguishability security, which we subsequently bootstrap all the way to full-fledged UC security via a sequence of transformations. Additionally, the constructions in [BLPV18] assume that the starting OT protocol satisfies *both* receiver and sender oblivious sampleability, while our starting r-iOT protocol is *not* required to satisfy sender oblivious sampleability.

Main Results. Figure 1 summarizes the main results of this paper. Our first main result is a generic construction of UC-secure two-round adaptive MPC from any r-iOT protocol. In somewhat more detail, our first result can be summarized as follows:

Theorem 1 (Informal). *Assuming* r-iOT*, i.e, a two-message OT protocol that satisfies indistinguishability security and receiver oblivious sampleability against static corruption of the sender/receiver by malicious adversaries in the CRS model (resp. semi-honest adversaries in the plain model), there exists a two-round MPC protocol for any functionality f that satisfies UC security against adaptive corruption of any subset of the parties by malicious adversaries in the CRS model (resp. semi-honest adversaries in the plain model).*

We achieve this result via a sequence of transformations that build progressively stronger OT protocols from weaker ones. These transformations use

a number of additional cryptographic primitives, all of which we show can be built in a generic way from any r-iOT protocol in the appropriate model.

Next, we show how to instantiate an r-iOT protocol in various models from a variety of concrete assumptions, including CDH, LPN, and isogeny-based assumptions. In somewhat more details, our second main result can be summarized as follows:

Theorem 2 (Informal). *Assuming CDH or LPN, there exists a construction of* r-iOT *that is secure against malicious adversaries in the CRS model (resp. semi-honest adversaries in the plain model).*

Theorem 3 (Informal). *Under certain isogeny-based assumptions (notably, CSIDH [CLM+18] or CSI-FiSh [BKV19]), there exists a construction of* r-iOT *that is secure against malicious adversaries in the CRS model.*

Our constructions of r-iOT from CDH and LPN build upon previous work due to Döttling *et al.* [DGH+20] that realized UC-secure OT/MPC against static corruptions from the same set of assumptions. Our construction of r-iOT from isogeny-based assumptions is based on a novel usage of the *(restricted) effective group action* framework due to Alamati *et al.* [ADMP20]. In particular, we show how to use a trusted setup to bypass issues around sampling obliviously from the "set" of an effective group action, which is a well-known open problem in the isogeny literature [Pet17, DMPS19, CPV20].[7]

Combined with the previous theorem, we obtain as a corollary the *first* constructions of two-round adaptively UC-secure MPC against malicious adversaries from the same concrete assumptions:

Corollary 1 (Informal). *Assuming CDH, LPN, or certain isogeny-based assumptions (notably, CSIDH [CLM+18] or CSI-FiSh [BKV19]), there exists a two-round MPC protocol for any functionality f that satisfies UC security against adaptive corruption of any subset of the parties by malicious adversaries in the CRS model.*

In summary, we show that it is feasible to construct round-optimal maliciously secure MPC in the adaptive corruption model from essentially all of the commonly used cryptographic assumptions. This essentially closes the gap between the static corruption model and the adaptive corruption model in terms of constructing round-optimal maliciously secure MPC from concrete assumptions. Figure 2 presents a high-level summary of our roadmap from r-iOT to adaptively UC-secure MPC.

[7] Unlike CDH or LPN, we do not achieve a construction of r-iOT from isogeny-based assumptions in the plain model. Achieving this seemingly requires new techniques for sampling obliviously from the "set" of an effective group action beyond those used in state-of-the-art isogeny-based cryptography.

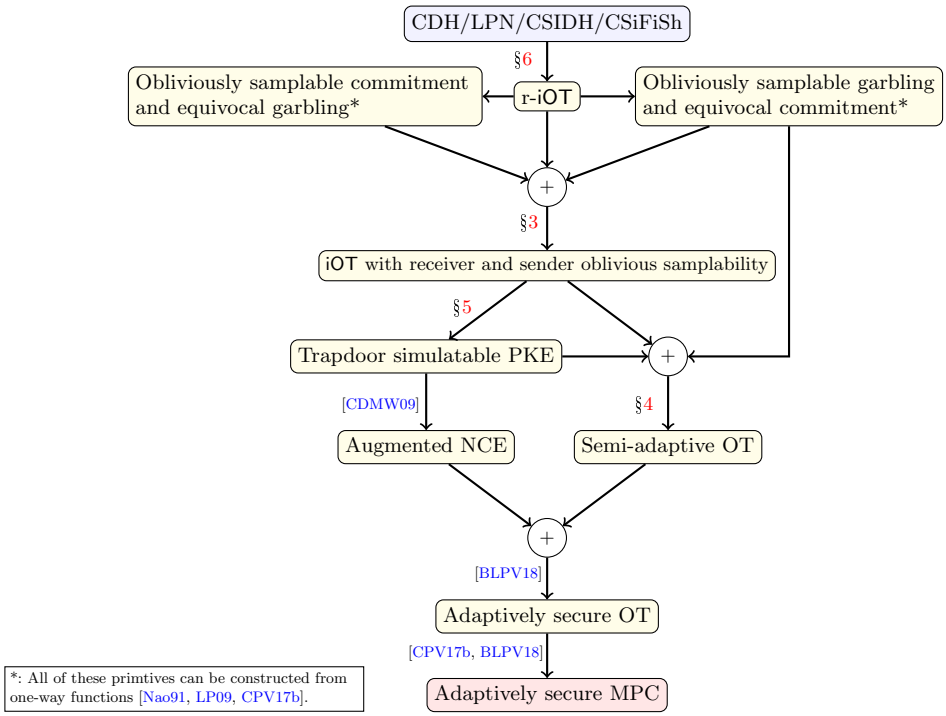

Fig. 2. An overview of our results

Additional Results. Besides our main contributions, we show some additional results that could be of independent interest. In particular, we show that any r-iOT protocol that is secure against semi-honest adversaries implies the existence of a *trapdoor-simulatable* PKE, which in turn is known to imply *non-committing encryption* (NCE) in a generic manner (in fact, it was shown in [CDMW09] that trapdoor-simulatable PKE implies an "augmented" variant of NCE). Due to its wide range of applications, NCE (and its augmented variants) have been studied by a long line of works [CFGN96, CDMW09, CPR17, YKT19, BBD+20].

Theorem 4 (Informal). *Any* r-iOT *that is secure against semi-honest adversaries implies a construction of trapdoor-simulatable PKE.*

Combined with the previous theorem on instantiations of r-iOT from concrete assumptions, and the known implication due to [CDMW09], we obtain as a corollary the *first* constructions (to our knowledge) of (augmented) NCE from LPN or certain isogeny-based assumptions:

Corollary 2 (Informal). *Assuming LPN, there exists a construction of a two round (augmented) non-committing encryption (NCE) scheme.*

Corollary 3 (Informal). *Assuming isogeny-based assumptions such as CSIDH [CLM+18] or CSI-FiSh [BKV19], there exists a construction of a two round (augmented) non-committing encryption (NCE) scheme in the CRS model.*

Complexity Analysis of Our Constructions. Our constructions may be viewed primarily as feasibility results for adaptive OT/MPC, and hence they are not tuned for practical efficiency. For the sake of completeness, we present here an asymptotic complexity analysis of the number of public-key operations used in our constructions. We assume that all messages are κ-bit, where κ is the security parameter. We also assume that the commitment schemes underlying our constructions require $O(\kappa)$ bits of randomness to commit to a bit.

1. Our construction of bit iOT in Section 3 requires $O(\kappa)$ executions of the underlying $\mathsf{r\text{-}iOT}$ protocol.
2. Our construction of trapdoor-simulatable PKE in Section 5 requires $O(1)$ executions of the underlying $\mathsf{r\text{-}iOT}$ protocol.
3. Our construction of semi-adaptive OT in Section 4 requires $O(\kappa)$ executions of both the underlying (string) iOT protocol as well as the underlying trapdoor-simulatable PKE scheme.
4. The construction of augmented NCE in [CDMW09] requires $O(\kappa)$ executions of a trapdoor-simulatable PKE scheme.
5. The construction of adaptive OT in [BLPV18] requires $O(1)$ executions of both the underlying semi-adaptive OT protocol and the underlying augmented NCE scheme.

Thus, asymptotically, the construction of adaptive OT based on our proposed framework requires $\mathcal{O}(\kappa^2)$ executions of the bit iOT protocol. This translates to $\mathcal{O}(\kappa^3)$ executions of the underlying $\mathsf{r\text{-}iOT}$ protocol for a $\mathcal{O}(\kappa)$-bit message. Finally, we analyze the number of public-key operations required in the various instantiations of $\mathsf{r\text{-}iOT}$ (for $O(\kappa)$-bit messages) from concrete assumptions:

- The construction of $\mathsf{r\text{-}iOT}$ from CDH (resp., LPN) assumption in [DGH+20] requires $O(\kappa)$ exponentiation operations (resp., LPN-sample generations).
- Our construction of $\mathsf{r\text{-}iOT}$ from isogeny-based assumptions (more concretely, from restricted effective group actions) in Section 6.1 requires $O(\kappa\ell)$ group action computations for any $\ell = \omega(\log \kappa)$.

Outline. The rest of the paper is organized as follows. Section 2 presents notations and definitions for two-round OT protocols in the CRS model. Section 3 describes our construction of two-round iOT with both receiver and sender oblivious sampleability from any two-round $\mathsf{r\text{-}iOT}$ protocol. Section 4 describes our construction (and proof) of semi-adaptively secure two-round OT from any two-round iOT with both receiver and sender oblivious sampleability. Section 5 presents our construction of trapdoor simulatable PKE (and augmented NCE) from any two-round $\mathsf{r\text{-}iOT}$ protocol. Section 6 describes our concrete constructions of two-round $\mathsf{r\text{-}iOT}$ from isogeny-based assumptions, CDH or LPN. Due to lack of space, we defer some additional background material and detailed proofs to the full version of our paper.

2 Preliminaries

In this section, we present some core preliminaries that are integral to our constructions. We defer many definitions and other background with which we expect most readers to be familiar to the full version of our paper.

2.1 Notations

We denote by $a \leftarrow D$ a uniform sampling of an element a from a distribution D. The set of elements $\{1, \ldots, n\}$ is represented by $[n]$. We denote polylog(a) and poly(b) as polynomials in $\log a$ and b respectively. We denote a probabilistic polynomial time algorithm as PPT. We denote the computational security parameter by κ. We denote a negligible function in κ as $\mathsf{neg}(\kappa)$. When a party S gets corrupted we denote it by S^*. Our security proofs are in the Universal Composability (UC) framework of [Can01]. We refer to the original paper for details. We denote computational and statistical indistinguishability by $\overset{c}{\approx}$ and $\overset{s}{\approx}$ respectively. We abbreviate "common reference string" as CRS. Unless otherwise specified, our constructions and proofs are in "local" CRS model. This happens to be the same CRS model in which the prior work due to Benhamouda *et al.* [BLPV18] showed constructions of adaptive MPC protocols with security against malicious adversaries.

2.2 Two-Message Oblivious Transfer in the CRS Model

In this section, we formally define a two-message oblivious transfer (OT) protocol in the common reference string (CRS) model. We then define two security notions for such an OT protocol, namely universal composability (UC) security and a weaker notion of indistinguishability-based security. We first focus on security against static corruptions by a malicious adversary. Subsequently, we discuss different levels of adaptive security.

A two-message OT protocol in the CRS model is a tuple of four algorithms of the form $\mathsf{OT} = (\mathsf{Setup}, \mathsf{OTR}_1, \mathsf{OTS}, \mathsf{OTR}_2)$ described below:

- $\mathsf{Setup}(1^\kappa)$: Takes as input the security parameter κ and outputs a CRS string crs and a trapdoor td. [8]
- $\mathsf{OTR}_1(\mathsf{crs}, b \in \{0,1\})$: Takes as input the crs and a bit $b \in \{0,1\}$, and outputs the receiver's message $\mathsf{M_R}$ and the receiver's internal state st.
- $\mathsf{OTS}(\mathsf{crs}, \mathsf{M_R}, m_0, m_1)$: Takes as input the crs, the receiver's message $\mathsf{M_R}$, a pair of input strings (m_0, m_1), and outputs the sender's message $\mathsf{M_S}$.
- $\mathsf{OTR}_2(\mathsf{crs}, \mathsf{M_S}, b, \mathsf{st})$: Takes as input the crs, the sender's message $\mathsf{M_S}$, a bit b, and receiver's internal state st, and outputs a message string m'.

[8] For standard two-message OT protocols, the setup algorithm need not output a trapdoor td, but we include it for certain security properties described subsequently.

Correctness. A two-message OT protocol in the CRS model is said to be correct if for any $b \in \{0,1\}$ and any (m_0, m_1), letting $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\kappa)$ and $(\mathsf{M_R}, \mathsf{st}) \leftarrow \mathsf{OTR}_1(\mathsf{crs}, b)$, the following holds with overwhelming probability:

$$\mathsf{OTR}_2(\mathsf{crs}, \mathsf{OTS}(\mathsf{crs}, \mathsf{M_R}, m_0, m_1), b, \mathsf{st}) = m_b.$$

Corruption Models. We consider the following (progressively non-decreasing in strength) adversarial models against any two-message OT protocol:

- *Static Corruption:* The adversary corrupts the parties at the onset of the protocol.
- *Semi-Adaptive Corruption:* The adversary corrupts one party (either the receiver or the sender) adaptively (at any point before/during/after the protocol) and the other party statically at the beginning of the protocol.
- *Adaptive Corruption:* The adversary corrupts both parties adaptively (at any point before/during/after the protocol). This scenario covers the previous corruption cases.

Indistinguishability-Based Security. We also consider a weaker notion of indistinguishability-based security against malicious adversaries in the static corruption setting. This notion is adopted directly from [DGH+20]. A two-message OT protocol $\mathsf{iOT} = (\mathsf{Setup}, \mathsf{iOTR}_1, \mathsf{iOTS}, \mathsf{iOTR}_2)$ satisfies indistinguishability-based security if the following properties hold:

Receiver's Indistinguishability Security. Formally, receiver's indistinguishability security requires that the following holds for any $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\kappa)$:

$$(\mathsf{crs}, \mathsf{iOTR}_1(\mathsf{crs}, 0)) \overset{c}{\approx} (\mathsf{crs}, \mathsf{iOTR}_1(\mathsf{crs}, 1)).$$

Sender's Indistinguishability Security. Sender's indistinguishability security is defined in [DGH+20] via an experiment $\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},r,w,b}(\mathcal{A})$ between a non-uniform PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a challenger, where the experiment is parameterized by some honestly generated crs, random coins $r \in \{0,1\}^\kappa$, an integer n representing the bitwise length of messages, a bit $w \in \{0,1\}$, and a bit $b \in \{0,1\}$:

$\underline{\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},r,w,b}(\mathcal{A})}$:

1. Run $(m_0, m_1, \mathsf{M_R}, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{crs})$.
2. If $b = 0$, compute $\mathsf{M_S} \leftarrow \mathsf{iOTS}(\mathsf{crs}, \mathsf{M_R}, (m_0, m_1))$.
3. If $b = 1$, compute $\mathsf{M_S} \leftarrow \mathsf{iOTS}(\mathsf{crs}, \mathsf{M_R}, (m'_0, m'_1))$ where $m'_w \leftarrow \{0,1\}^n$ and $m'_{1-w} := m_{1-w}$.
4. Output $s \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{M_S})$.

For a given $(\mathsf{crs}, r, w \in \{0,1\})$, we define the advantage $\mathcal{A}_{\mathsf{iOT}}^{\mathsf{crs},r,w}(\mathcal{A})$ as:

$$\mathsf{Adv}_{\mathsf{iOT}}^{\mathsf{crs},r,w}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},r,w,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},r,w,1}(\mathcal{A}) = 1].$$

We say that iOT satisfies sender's indistinguishability security if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{crs},r,w}_{\mathsf{iOT}}(\mathcal{A})$ is negligible in κ for at least one $w \in \{0,1\}$, where the probability is taken over $\mathsf{crs} = \mathsf{Setup}(1^\kappa)$ and $r \leftarrow \{0,1\}^\kappa$.[9]

2.3 iOT with Oblivious Sampleability

We also consider notions of *oblivious sampleability* for indistinguishability-secure two-message OT protocols in the CRS model. An iOT protocol of the form $\mathsf{iOT} = (\mathsf{Setup}, \mathsf{iOTR}_1, \mathsf{iOTS}, \mathsf{iOTR}_2)$ is said to satisfy oblivious sampleability if it supports additional "oblivious sampling" algorithms - $(\widetilde{\mathsf{iOTR}}, \widetilde{\mathsf{iOTS}})$ and the corresponding "randomness inversion" algorithms - $(\widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}, \widetilde{\mathsf{iOTS}_{\mathsf{Inv}}})$ defined as:

- $\widetilde{\mathsf{iOTR}}(\mathsf{crs}; r)$: Outputs an obliviously sampled receiver's message $\mathsf{M_R}$.
- $\widetilde{\mathsf{iOTS}}(\mathsf{crs}, w, m_{1-w}; r)$ Outputs an obliviously sampled sender's message $\mathsf{M_S}$.
- $\widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}(\mathsf{crs}, \mathsf{M_R}, \mathsf{td}, r)$: Outputs randomness $\tilde{r}$ corresponding to an honestly generated receiver message $\mathsf{M_R}$.
- $\widetilde{\mathsf{iOTS}_{\mathsf{Inv}}}(\mathsf{crs}, w, \mathsf{M_S}, \mathsf{td}, r)$: Outputs randomness $\tilde{r}$ corresponding to an honestly generated sender message $\mathsf{M_S}$.

We say that the iOT is obliviously sampleable if it satisfies *both* receiver and sender oblivious sampleability, as defined below.

Receiver Oblivious Sampleability: For any bit $b \in \{0,1\}$, an obliviously sampled receiver's message should be indistinguishable from an honestly generated one, even given the sampling randomness. More formally, we require that for any $(\mathsf{crs}, \mathsf{td}) = \mathsf{Setup}(1^\kappa)$ and any bit $b \in \{0,1\}$, we have $(\mathsf{crs}, \mathsf{M_R}, \widehat{r}) \overset{c}{\approx} (\mathsf{crs}, \widetilde{\mathsf{M_R}}, \widetilde{r})$, where for uniformly random coins $r, \widetilde{r} \leftarrow \{0,1\}^\kappa$, we have

$$\mathsf{M_R} = \mathsf{iOTR}(\mathsf{crs}, b; r), \quad \widehat{r} = \widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}(\mathsf{crs}, \mathsf{M_R}, \mathsf{td}, r), \quad \widetilde{\mathsf{M_R}} = \mathsf{iOTR}(\mathsf{crs}, b; \widetilde{r}).$$

Sender Oblivious Sampleability: We also require that a corrupt receiver cannot infer whether the sender's message (corresponding to the bit w which is not chosen by the receiver) was obliviously sampled or generated honestly. We consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ participating in an experiment $\mathsf{Exp}^{\mathsf{crs},r,w,b}_{\widetilde{\mathsf{iOT}}}(\mathcal{A})$, indexed by a crs, random coins $r \in \{0,1\}^\kappa$, a bit $w \in \{0,1\}$ and a bit $b \in \{0,1\}$:

$\underline{\mathsf{Exp}^{\mathsf{crs},r,w,b}_{\widetilde{\mathsf{iOT}}}(\mathcal{A})}$:

- Run $(m_0, m_1, \mathsf{M_R}, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{crs}; r)$.
- If $b = 0$, sample randomness $\tilde{r}$ and compute $\mathsf{M_S} \leftarrow \widetilde{\mathsf{iOTS}}(\mathsf{crs}, w, m_{1-w}; \tilde{r})$.

[9] This is slightly different from the traditional notion of sender's indistinguishability security for two-message OT; we refer to [DGH+20] for more details.

– If $b = 1$, sample randomness $\widehat{r}$, compute $\mathsf{M_S} \leftarrow \mathsf{iOTS}(\mathsf{crs}, \mathsf{M_R}, (m_0, m_1); \widehat{r})$ and $\tilde{r} = \widetilde{\mathsf{iOTS}}_{\mathsf{Inv}}(\mathsf{crs}, w, \mathsf{M_S}, \mathsf{td}, \widehat{r})$.
– Compute and output $s \leftarrow \mathcal{A}_2(\mathsf{st}, \tilde{r}, \mathsf{M_S})$.

Define the advantage of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{crs},r,w}_{\widetilde{\mathsf{iOT}}}(\mathcal{A}) = |\Pr[\mathsf{Exp}^{\mathsf{crs},r,w,0}_{\widetilde{\mathsf{iOT}}}(\mathcal{A}) = 1] - \Pr[\mathsf{Exp}^{\mathsf{crs},r,w,1}_{\widetilde{\mathsf{iOT}}}(\mathcal{A}) = 1]|.$$

We say that iOT satisfies sender oblivious sampleability if for any PPT adversary $\mathcal{A}$ and any $w \in \{0,1\}$, $\mathsf{Adv}^{\mathsf{crs},r,w}_{\widetilde{\mathsf{iOT}}}(\mathcal{A})$ is negligible in κ, where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\kappa)$ and $r \leftarrow \{0,1\}^\kappa$.

r-iOT. We denote by $\mathsf{r\text{-}iOT} = (\mathsf{Setup}, \mathsf{r\text{-}iOTR}_1, \mathsf{r\text{-}iOTS}, \mathsf{r\text{-}iOTR}_2, \widetilde{\mathsf{r\text{-}iOTR}}, \widetilde{\mathsf{r\text{-}iOTR}}_{\mathsf{Inv}})$, an indistinguishability-secure two-message OT in the CRS model that satisfies receiver oblivious sampleability but *not necessarily* sender oblivious sampleability. Such an OT protocol only needs to support an "oblivious sampling" algorithm $\widetilde{\mathsf{r\text{-}iOTR}}$ (where $\widetilde{\mathsf{r\text{-}iOTR}}$ is defined similar to iOTR) and the corresponding "randomness inversion" algorithm $\widetilde{\mathsf{r\text{-}iOTR}}_{\mathsf{Inv}}$ (where $\widetilde{\mathsf{r\text{-}iOTR}}_{\mathsf{Inv}}$ is defined similar to $\widetilde{\mathsf{iOTR}}_{\mathsf{Inv}}$) for the receiver.

2.4 Garbling Schemes

A garbling scheme [Yao86, CPV17b] is a tuple $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev})$, described as follows:

– $\mathsf{Gb}\,(1^\kappa, \mathcal{C}) \to (\mathrm{GC}, \mathsf{Keys})$: A randomized algorithm which takes as input the security parameter and a circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^m$ and outputs a tuple of strings $(\mathrm{GC}, \mathsf{Keys})$, where GC is the garbled circuit and Keys denotes the input-wire labels.
– $\mathsf{En}\,(x, \mathsf{Keys}) = \mathsf{X}$: a deterministic algorithm that outputs the garbled input X corresponding to input x.
– $\mathsf{Ev}\,(\mathrm{GC}, \mathsf{X}) = y$: A deterministic algorithm which evaluates garbled circuit GC on garbled input X and outputs y.

We borrow this definition and the associated notations from the work of [CPV17b]. The garbling scheme used in our protocols needs to satisfy standard properties such as correctness and privacy (we refer to [Yao86, CPV17b] for the definitions). We additionally borrow two extra properties from [CPV17b] for our garbling schemes: namely, oblivious sampleability and equivocability, which we define here.

Oblivious Sampleability. Oblivious sampleability allows the garbler to obliviously sample a garbled circuit without the knowledge of the input keys Keys. It also enables an honestly computed garbled circuit to be claimed as obliviously sampled. A garbling scheme $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev})$ is said to satisfy oblivious sampleability if there exist PPT algorithms $\widetilde{\mathsf{Gb}}$ and $\mathsf{Gb}_{\mathsf{Inv}}$ defined as:

- $\widetilde{\mathsf{Gb}}(1^\kappa, \mathcal{C}, y) \to (\widetilde{\mathrm{GC}}, \widetilde{\mathsf{X}})$: A randomized algorithm that outputs an obliviously sampled garbled circuit $\widetilde{\mathrm{GC}}$ and obliviously sampled wire labels $\widetilde{\mathsf{X}}$ such that evaluating $\widetilde{\mathrm{GC}}$ on $\widetilde{\mathsf{X}}$ would yield y as output,
- $\mathsf{Gb}_{\mathsf{Inv}}(r, \mathsf{Keys}, x) \to \widehat{r}$: A randomness inversion algorithm that given some randomness of garbling r, input-wire labels Keys, and an input x, outputs some random coins $\widehat{r}$,

such that for any polynomial-time circuit $\mathcal{C}$ and for all input output pairs (x, y) such that $\mathcal{C}(x) = y$ it holds that

$$(\mathsf{Gb}_{\mathsf{Inv}}(r, \mathsf{Keys}, x), \mathrm{GC}, \mathsf{X}) \overset{c}{\approx} (\widetilde{r}, \widetilde{\mathrm{GC}}, \widetilde{\mathsf{X}}),$$

where for random coins $r, \widetilde{r} \leftarrow \{0,1\}^\kappa$, we have

$$(\mathrm{GC}, \mathsf{Keys}) = \mathsf{Gb}(1^\kappa, \mathcal{C}; r), \quad \mathsf{X} = \mathsf{En}(x, \mathsf{Keys}), \quad (\widetilde{\mathrm{GC}}, \widetilde{\mathsf{X}}) = \widetilde{\mathsf{Gb}}(1^\kappa, \mathcal{C}, y; \widetilde{r}).$$

Equivocal Garbling. Finally, we require the garbled circuit to be equivocal [CPV17b]. It allows a privacy simulator $\mathcal{S}_{\mathrm{GC}}$ to generate a fake garbled circuit $\widetilde{\mathrm{GC}}$ and fake input wire labels $\widetilde{\mathsf{X}}$ that always evaluate to a fixed output. Later, the simulator can open $(\widetilde{\mathrm{GC}}, \widetilde{\mathsf{X}})$ to a particular input x by providing consistent randomness used in the garbling process. We define this as follows: a garbling scheme $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \widetilde{\mathsf{Gb}})$ is said to be equivocal if there exists a pair of PPT algorithms $(\mathcal{S}^1_{\mathrm{GC}}, \mathcal{S}^2_{\mathrm{GC}})$, such that any PPT adversary $\mathcal{A}$ wins the following game with at most negligible advantage:

1. $\mathcal{A}$ gives a circuit $\mathcal{C}$ and an input x to the challenger.
2. The challenger flips a bit b.
   - If $b = 0$: It computes $(\mathrm{GC}, \mathsf{Keys}) \leftarrow \mathsf{Gb}(\mathcal{C}; r)$ and $\mathsf{X} \leftarrow \mathsf{En}(x, \mathsf{Keys})$. It sends $\mathrm{GC}, \mathsf{X}, \mathsf{Keys}, r$ to the adversary $\mathcal{A}$.
   - If $b = 1$: It sets $y = C(x)$. It runs the simulator $(\mathrm{GC}, \mathsf{X}, \mathsf{st}) \leftarrow \mathcal{S}^1_{\mathrm{GC}}(\mathcal{C}, y)$. It runs the simulator $(\mathsf{Keys}, r) \leftarrow \mathcal{S}^2_{\mathrm{GC}}(\mathsf{st}, x)$. It sends $\mathrm{GC}, \mathsf{X}, \mathsf{Keys}, r$ to the adversary $\mathcal{A}$.
3. The adversary outputs a bit b'.

The adversary wins if $b = b'$.

2.5 Additional Cryptographic Primitives

In this section, we define certain additional cryptographic primitives that we require for our constructions: namely, equivocal commitments with oblivious sampleability, and trapdoor simulatable PKE.

Equivocal Commitment. Let $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{Ver}, \mathsf{Equiv})$ be an equivocal commitment scheme in the CRS model as defined in [BLPV18]. We say that Com is *obliviously sampleable* if there exist additional algorithms $(\widetilde{\mathsf{Com}}, \widetilde{\mathsf{Com}}_{\mathsf{Inv}})$ for oblivious commitment generation and randomness inversion, respectively, such that for any $(\mathsf{crs}, \mathsf{td}) = \mathsf{Setup}(1^\kappa)$ and any message m, we have $(c, \widehat{r}) \overset{c}{\approx} (\widetilde{c}, \widetilde{r})$, where

$$c = \mathsf{Com}(\mathsf{crs}, m; r), \quad \widehat{r} = \widetilde{\mathsf{Com}}_{\mathsf{Inv}}(\mathsf{crs}, m, r, \mathsf{td}), \quad \widetilde{c} = \widetilde{\mathsf{Com}}(\mathsf{crs}; \widetilde{r}),$$

for random coins $r, \widetilde{r} \leftarrow \{0,1\}^\kappa$. Such an equivocal commitment scheme with oblivious sampleability can be obtained from one-way-functions [Nao91].

Trapdoor Simulatable PKE. We recall the definition of trapdoor simulatable PKE from [CDMW09]. A trapdoor simulatable PKE scheme in the CRS model is a tuple of the form $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{oGen}, \mathsf{oEnc})$, where the tuple $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a standard PKE scheme that is augmented with oblivious sampling algorithms $(\mathsf{oGen}, \mathsf{oEnc})$ and randomness inverting algorithms $(\mathsf{rGen}, \mathsf{rEnc})$. The trapdoor of the setup string allows generating a public key (resp. a ciphertext) honestly and then claiming that the public key (resp. a ciphertext) was obliviously sampled using the rGen (resp. the rEnc) algorithm. Formally, we require that for any message $m \in \{0,1\}^\ell$, letting $(\mathsf{crs}, \mathsf{td}) = \mathsf{Setup}(1^\kappa)$,

$$(\mathsf{pk}, c, \widehat{r}_G, \widehat{r}_E) \overset{c}{\approx} (\widetilde{\mathsf{pk}}, \widetilde{c}, \widetilde{r}_G, \widetilde{r}_E),$$

where for random coins $r_G, r_E, \widetilde{r}_G, \widetilde{r}_E \leftarrow \{0,1\}^\kappa$, we have $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(\mathsf{crs}; r_G)$, $c = \mathsf{Enc}(\mathsf{crs}, \mathsf{pk}, m; r_E)$, $\widetilde{\mathsf{pk}} = \mathsf{oGen}(\mathsf{crs}; \widetilde{r}_G)$, $\widetilde{c} = \mathsf{oEnc}(\mathsf{crs}; \widetilde{r}_E)$, and

$$\widehat{r}_G = \mathsf{rGen}(\mathsf{crs}, r_G, \mathsf{td}), \quad \widehat{r}_E = \mathsf{rEnc}(\mathsf{crs}, r_G, r_E, m, \mathsf{td}).$$

3 iOT with Oblivious Sampleability from r-iOT

In this section, we present the first generic construction in our overall framework: we show how to build a two-message iOT protocol in the CRS model with *both* oblivious sender and receiver sampleability given a two-message $\mathsf{r\text{-}iOT}$ protocol (iOT with receiver oblivious sampleability but *not necessarily* sender oblivious sampleability). For simplicity of exposition, we describe the construction in the CRS model against malicious corruptions; the corresponding construction in the plain model against semi-honest corruptions follows analogously.

3.1 Construction Overview and Intuition

We construct a two-message iOT protocol with oblivious sender and receiver sampleability in the CRS model given the following ingredients: (1) a two-message $\mathsf{r\text{-}iOT}$ protocol in the CRS model, (2) an equivocal garbling scheme, (3) an obliviously sampleable garbling scheme, and (4) an obliviously sampleable commitment scheme (in the CRS model). The latter three schemes are implied by one-way functions (and hence by $\mathsf{r\text{-}iOT}$).

A First Attempt. We describe below an initial attempt to build iOT with receiver and sender oblivious sampleability from $\mathsf{r\text{-}iOT}$. This simple construction additionally uses a standard garbling scheme and a standard (non-interactive) commitment scheme. Additionally, let $\mathcal{C}[\beta, c](\cdot, \cdot)$ denote a circuit that is hardwired with a bit $\beta \in \{0,1\}$ and a commitment c. It takes as input some randomness r and a message m, and outputs m if c is valid commitment to β using randomness r. Otherwise, it outputs $\perp$ (the circuit $\mathcal{C}$ is also hardwired with the CRS string for the commitment scheme, but we avoid mentioning this explicitly for simplicity of presentation.).

– iOTR_1: The receiver uses the commitment scheme to create a commitment c to its input choice bit b under randomness r. The receiver transmits this commitment c to the sender. The receiver also uses the underlying r-iOT protocol to send one r-iOT-receiver message corresponding to each bit of the randomness r (in parallel).
– iOTS: The sender uses the commitment c from the receiver to create two circuits $\mathcal{C}_{0,c}(\cdot, m_0)$ and $\mathcal{C}_{1,c}(\cdot, m_1)$ as described earlier. It then garbles these circuits using the garbling scheme to create a pair of garbled circuits GC_0 and GC_1, along with the corresponding wire labels for their input bits.

  The sender sends across GC_0 and GC_1 to the receiver. In parallel, the sender uses the underlying r-iOT scheme and the r-iOT messages from the receiver to generate one r-iOT-sender message for each pair of wire labels, and also sends all of these across to the receiver.
– iOTR_2: The receiver uses the r-iOT-sender messages to recover the wire labels corresponding to its randomness string r for both garbled circuits GC_0 and GC_1. It then evaluates GC_b on r by using the corresponding wire labels to recover the message m_b.

The above approach fails to give us sender's oblivious sampleability. Note that the sender's message has two parts - the garbled circuits $(\mathrm{GC}_0, \mathrm{GC}_1)$, and the r-iOT-sender messages. Using an *obliviously sampleable garbling scheme* naturally allows oblivious sampleability for the first part of the sender's message. However, it is not clear if the second part of the sender's message can be obliviously sampled since the underlying r-iOT protocol does not necessarily support sender oblivious sampleability.

Our Solution. We address this issue by using *two separate sets* of garbled circuits. The first set of garbled circuits are created using a garbling scheme Garble' that is *obliviously sampleable* [LP09], while the second set of garbled circuits are created using a garbling scheme Garble that is *equivocal* [CPV17b]. The complete solution is described formally in Figure 3. The oblivious sampling and randomness inversion algorithms are described in Figure 4.

Theorem 5. *Assuming that: (1) $\pi_{\mathsf{r\text{-}iOT}}$ is a two-message r-iOT protocol in the $\mathsf{crs}_{\mathsf{iOT}}$ model, (2) Com is an obliviously sampleable commitment scheme, (3)* Garble *is an equivocal garbling scheme, and (4)* Garble' *is an obliviously sampleable garbling scheme, π_{iOT} is a two-message iOT protocol with sender and receiver oblivious sampleability in the CRS model.*

The formal security proof is deferred to the full version of the paper. We present here a high-level overview of the arguments for indistinguishability security and oblivious sampleability (for both sender and receiver) of our protocol.

Indistinguishability Security (Informal). Arguing receiver's indistinguishability security is again straightforward. Informally, the commitment c computationally hides the receiver's choice bit b because the r-iOT messages sent by the receiver computationally hide the receiver's randomness string r.

Fig. 3. Constructing iOT with Oblivious Sampleability from $\mathsf{r\text{-}iOT}$

π_{iOT}

- **Public Inputs:** $\mathsf{crs}_{\mathsf{iOT}} = (\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \mathsf{crs}_{\mathsf{com}})$ where $\mathsf{crs}_{\mathsf{r\text{-}iOT}}$ and $\mathsf{crs}_{\mathsf{com}}$ are the setup strings of $\mathsf{r\text{-}iOT}$ and Com respectively.
- **Circuits:** Circuit $\mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, \beta](r, p) = p$ if $c = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, \beta; r)$, else $\mathcal{C}$ outputs $\perp$. Circuit $\mathcal{C}'[\mathsf{crs}_{\mathsf{com}}](c', s, m) = m$ if $c' = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, 0; s)$, else it outputs $\perp$.
- **Private Inputs:** S has input bits (m_0, m_1) where $m_0, m_1 \in \{0,1\}$; R has input choice bit b.
- **Primitives:** Let $\pi_{\mathsf{r\text{-}iOT}} = (\mathsf{r\text{-}iOTR}_1, \mathsf{r\text{-}iOTS}, \mathsf{r\text{-}iOTR}_2, \widetilde{\mathsf{r\text{-}iOTR}})$ denote a receiver obliviously sampleable indistinguishable OT. $(\mathsf{Com}, \widetilde{\mathsf{Com}})$ is an obliviously sampleable commitment scheme. $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \mathcal{S}_{\mathsf{GC}})$ is an equivocal garbling scheme. $\mathsf{Garble}' = (\mathsf{Gb}', \mathsf{En}', \mathsf{Ev}', \widetilde{\mathsf{Gb}}')$ is an obliviously sampleable garbling scheme.

$\mathbf{iOTR}_1(\mathsf{crs}_{\mathsf{iOT}}, b)$:

- R commits to b using randomness r as $c = \mathsf{Com}(b; r)$. Let $|r| = \ell$.
- R computes $\pi_{\mathsf{r\text{-}iOT}}$ receiver messages as $\{\gamma_{0,i}, \gamma_{1,i}\}$ where $\gamma_{b,i} = \mathsf{r\text{-}iOTR}_1(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, r_i)$ and $\gamma_{1-b,i} \leftarrow \widetilde{\mathsf{r\text{-}iOTR}}(\mathsf{crs}_{\mathsf{r\text{-}iOT}})$ for $i \in [\ell]$.
- R sends $\mathsf{M}_{\mathsf{R}} = \big(c, \{\gamma_{0,i}, \gamma_{1,i}\}_{i\in[\ell]}\big)$ as the receiver's message.

$\mathbf{iOTS}(\mathsf{crs}_{\mathsf{iOT}}, \mathsf{M}_{\mathsf{R}}, (m_0, m_1))$:
S runs the following algorithm for $\beta \in \{0,1\}$:

- S computes $c'_\beta = \mathsf{Com}(0; s_\beta)$. S garbles $\big(\mathrm{GC}'_\beta, \mathsf{Keys}'_\beta\big) = \mathsf{Gb}'(1^\kappa, \mathcal{C}'[\mathsf{crs}_{\mathsf{com}}])$. S computes $\{\mathsf{Y}_{\beta,i}\}_{i\in[t+\ell+1]} = \mathsf{En}'(c'_\beta || s_\beta || m_\beta, \mathsf{Keys}'_\beta)$.
- S sets $p_\beta = \{\mathsf{Y}_{\beta,i}\}_{i\in[t]}$ as the garbled input for c'_β corresponding to GC'_β. Let $|p_\beta| = t\kappa$. S garbles another garbled circuit for circuit $\mathcal{C}$ as $\big(\mathrm{GC}_\beta, \mathsf{Keys}_\beta\big) = \big(\mathrm{GC}_\beta, \{\mathsf{X}^0_{\beta,i}, \mathsf{X}^1_{\beta,i}\}_{i\in[\ell+t\kappa]}\big) = \mathsf{Gb}(1^\kappa, \mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, \beta])$. Sender sets $\mathsf{X}^p_\beta = \mathsf{En}(p_\beta, \{\mathsf{X}^0_{\beta,i}, \mathsf{X}^1_{\beta,i}\}_{i\in[\ell,\ell+t\kappa]}, \mathsf{Keys}_\beta)$ as the garbled input for p_β corresponding to GC_β.
- S computes $\pi_{\mathsf{r\text{-}iOT}}$ sender messages for receiver's input r in GC_β as $\tau_{\beta,i} = \mathsf{r\text{-}iOTS}(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \gamma_{\beta,i}, (\mathsf{X}^0_{\beta,i}, \mathsf{X}^1_{\beta,1}))$ for $i \in [\ell]$.

S sends $\mathsf{M}_s = \{\mathsf{M}_{\mathsf{S},\beta}\}_{\beta\in\{0,1\}} = \{\mathrm{GC}_\beta, \mathrm{GC}'_\beta, \{\tau_{\beta,i}\}_{i\in[\ell]}, \mathsf{X}^p_\beta, \{\mathsf{Y}_{\beta,i}\}_{i\in[t,t+\ell]}\}_{\beta\in\{0,1\}}$.

$\mathbf{iOTR}_2(\mathsf{crs}_{\mathsf{iOT}}, \mathsf{M}_{\mathsf{S}}, b)$:

- R computes the wire labels corresponding to commitment randomness r in GC_b as $\mathsf{X}_i = \mathsf{r\text{-}iOTR}_2(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \tau_{b,i})$ for $i \in [\ell]$. R evaluates GC to receive garbled input U for c'_b corresponding to GC' - $\mathsf{U} = \mathsf{Ev}(\mathrm{GC}, \{\mathsf{X}_i\}_{i\in[\ell+t\kappa]}) = \mathsf{U}$ where $|\mathsf{U}| = t\kappa$.
- R sets $\{\mathsf{Y}_i\}_{i\in[t]} = \{\mathsf{U}_i\}_{i\in t}$ where U_i is the ith chunk of κ bits of U. R outputs $m_b = \mathsf{Ev}'(\mathrm{GC}', \{\mathsf{Y}_i\}_{i\in[t+\ell]})$.

To argue sender's indistinguishability security, we point out that the only information about m_{1-b} that the receiver could learn is from the garbled circuit GC'_{1-b}. However, the receiver cannot evaluate GC'_{1-b} to m_{1-b} unless it learns p_{1-b}. As long as the receiver does not learn p_{1-b}, m_{1-b} is computationally hidden by the privacy of the garbling scheme itself. To see why the receiver cannot

Fig. 4. Oblivious Sampling and Randomness Inversion Algorithms for iOT

$\widetilde{\mathsf{iOTR}}(\mathsf{crs}_{\mathsf{iOT}})$:
R computes $c \leftarrow \widetilde{\mathsf{Com}}(\mathsf{crs}_{\mathsf{com}}; r_{\mathsf{Com}})$ and $\gamma_{\beta,i} \leftarrow \widetilde{\mathsf{r\text{-}iOTR}}(\mathsf{crs}_{\mathsf{r\text{-}iOT}}; r_{\mathsf{r\text{-}iOT},\beta,i})$ for $\beta \in \{0,1\}, i \in [\ell]$. Set $r_{\mathsf{r\text{-}iOT}} = \{r_{\mathsf{r\text{-}iOT},\beta,i}\}_{\beta\in\{0,1\},i\in[\ell]}$. R returns $r_{\mathsf{R}} = (r_{\mathsf{Com}}||r_{\mathsf{r\text{-}iOT}})$ as the sampling randomness.

$\widetilde{\mathsf{iOTR}}_{\mathsf{Inv}}(\mathsf{crs}, \mathsf{M}_{\mathsf{R}}, \mathsf{td} = (\mathsf{td}_{\mathsf{Com}}, \mathsf{td}_{\mathsf{r\text{-}iOT}}), r_{\mathsf{R}})$
Denote $r_{\mathsf{R}} = (r_{\mathsf{Com}}||r_{\mathsf{r\text{-}iOT}}||b||r)$. Obtain $\widetilde{r_{\mathsf{Com}}} = \widetilde{\mathsf{Com}}_{\mathsf{Inv}}(\mathsf{crs}_{\mathsf{com}}, b, r_{\mathsf{Com}}, \mathsf{td}_{\mathsf{Com}})$ and $\widetilde{r_{\mathsf{r\text{-}iOT}}} = \{\widetilde{\mathsf{r\text{-}iOTR}}_{\mathsf{Inv}}(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \gamma_{\beta,i}, \mathsf{td}_{\mathsf{r\text{-}iOT}}, r_{\mathsf{r\text{-}iOT},\beta,i})\}_{\beta\in\{0,1\},i\in[\ell]}$. Return $\widetilde{r_{\mathsf{R}}} = (\widetilde{r_{\mathsf{Com}}}||\widetilde{r_{\mathsf{r\text{-}iOT}}})$ as the randomness.

$\widetilde{\mathsf{iOTS}}(\mathsf{crs}_{\mathsf{iOT}}, w, m_{1-w}; r_{\mathsf{S}})$:

- S computes $\mathsf{M}_{\mathsf{S},1-w}$ correctly corresponding to message m_{1-w} using randomness r_{1-w}.
- For obliviously sampling $\mathsf{M}_{\mathsf{S},w}$, S performs the following using randomness r_w: S samples $m_w \leftarrow \{0,1\}$ and obliviously samples garbled circuit as $\mathrm{GC}'_w = \widetilde{\mathsf{Gb}}(1^\kappa, \mathcal{C}'[\mathsf{crs}_{\mathsf{com}}], m_w; r_{w,\mathrm{GC}'})$. S but correctly garbles $(\mathrm{GC}_w, \mathsf{Keys}_w) = \mathsf{Gb}(1^\kappa, \mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, b]; r_{w,\mathrm{GC}})$. Sender samples $\{(\mathsf{X}^0_{w,i}, \mathsf{X}^1_{w,i})\}_{i\in[\ell]} \leftarrow \{0,1\}^*$ randomly and samples X^p_w and $\{\mathsf{Y}_{w,i}\}_{i\in[t,t+\ell]}$ randomly. S computes $\tau_{w,i}$ messages correctly. S sets $\mathsf{M}_{s,w} = (\mathrm{GC}_w, \mathrm{GC}'_w, \{\tau_{w,i}\}_{i\in[\ell]}, \mathsf{X}^p_w, \{\mathsf{Y}_{w,i}\}_{i\in[t,t+\ell]})$.
- Return $\mathsf{M}_{\mathsf{S}} = (\mathsf{M}_{\mathsf{S},0}, \mathsf{M}_{\mathsf{S},1})$ and randomness $r_{\mathsf{S}} = (r_0||r_1)$.

$\widetilde{\mathsf{iOTS}}_{\mathsf{Inv}}(\mathsf{crs}, w, \mathsf{M}_{\mathsf{S}}, \mathsf{td} = \bot, r_{\mathsf{S}})$

- Denote $r_{\mathsf{S}} = (r_0||r_1)$ and set $\widetilde{r_{1-w}} = r_{1-w}$.
- r_w contains the garbling randomness for GC_w, GC'_w and $\{\tau_{w,i}\}_{i\in[\ell]}$ as $r_w = (r_{w,\mathrm{GC}}||r_{w,\mathrm{GC}'}||r_{w,\mathsf{r\text{-}iOT}})$. Set $\widetilde{r_{w,\mathrm{GC}}} = r_{w,\mathrm{GC}}$, $\widetilde{r_{w,\mathrm{GC}'}} = \mathsf{Gb}_{\mathsf{Inv}}(r_{w,\mathrm{GC}'}, \mathsf{Keys}'_w, 0^{t+\ell+1})$, where Keys' is the encoding information of GC'. Compute $\widetilde{r_w} = (\widetilde{r_{w,\mathrm{GC}}}||\widetilde{r_{w,\mathrm{GC}'}}||r_{w,\mathsf{r\text{-}iOT}})$.
- Output $\widetilde{r_{\mathsf{S}}} = (\widetilde{r_0}||\widetilde{r_1})$ and claim that $\mathsf{X}^p_w, \{\mathsf{Y}_{w,i}\}_{i\in[t,t+\ell]}$ were randomly sampled.

learn anything about p_{1-b}, observe that the only information about p_{1-b} that the receiver could learn is from the garbled circuit GC_{1-b}. However, the receiver cannot evaluate GC_{1-b} to anything other than $\bot$ since: (1) it cannot prove that c is a commitment to $(1-b)$ under randomness r (this follows from the binding property of the commitment scheme), and (2) it cannot recover any input labels to GC_{1-b} other than those corresponding to r (due to the sender privacy of the underlying $\mathsf{r\text{-}iOT}$ protocol).

Oblivious Sampleability (Informal). Finally, we argue informally that our new construction satisfies *both* receiver *and* sender oblivious sampleability.

Receiver Oblivious Sampleability. Given that we are starting with an $\mathsf{r\text{-}iOT}$ protocol that already satisfies receiver oblivious sampleability, arguing receiver oblivious sampleability for our overall construction is straightforward as long as we use a commitment scheme that is obliviously sampleable (this motivates us to use an obliviously sampleable commitment scheme).

Sender Oblivious Sampleability. We now argue that the modified construction also achieves sender oblivious sampleability. To obliviously sample a sender message for the branch $w = (1-b)$, we garble GC_w as per the "real" garbling scheme, but obliviously sample GC'_w (recall that GC'_w is generated using an obliviously sampleable garbling scheme Garble'). The $\mathsf{r\text{-}iOT}$ messages are computed using the wire labels of GC_w. To demonstrate sampleability, the simulator in the security experiment simply discloses the randomness used in the entire process as the sampling randomness.

The corresponding inversion algorithm takes as input the randomness used for correctly constructing GC_w, GC'_w and $\mathsf{r\text{-}iOT}$ messages. The simulator can now rely on the oblivious sampleability of the garbling scheme Garble' to claim that GC'_w was, in fact, obliviously sampled. The randomness for the honestly generated $\mathsf{r\text{-}iOT}$ messages and GC_w is provided as the sampling randomness. This is indistinguishable from an obliviously sampled sender message since in both cases GC_w evaluates to $\perp$.

At this point, we rely on the equivocal property of the garbling scheme Garble to argue that these two cases are indistinguishable since the inputs of GC_w are predetermined from receiver's OT message. This holds true even when the sampling adversary gets all the input wire labels for GC_w from the $\mathsf{r\text{-}iOT}$ randomness. This is the fundamental reason why we added the extra "layer" of garbling to our protocol. In the formal proof, this argument is a bit more technically involved: we need to also rely on distinguisher dependent simulation techniques [JKKR17, DGH+20]. We refer to the full version of our paper for details.

4 Semi-Adaptive OT from iOT with Oblivious Sampleability

In this section, we show how to build a semi-adaptively simulation-secure two-message OT protocol starting from a two-message iOT protocol with both receiver and sender oblivious sampleability in the static corruption setting. Coupled with our first generic construction from Section 3, this completes our roadmap to semi-adaptively simulation-secure two-message OT protocol starting from a two-message $\mathsf{r\text{-}iOT}$ protocol.

Construction Overview. To generate the receiver OT message, the receiver uses the equivocal commitment scheme to create a commitment c to its choice bit b under some appropriately sampled randomness r. Next, the receiver creates a set of encryptions (e_0, e_1). We need two encryptions instead of one to enable semi-adaptive security (which is discussed later on). The encryption e_b encrypts the commitment randomness r under the trapdoor simulatable PKE scheme using some appropriately sampled randomness s (we explain the intuition for this step subsequently). Meanwhile, $e_{\bar{b}}$ is obliviously sampled. The receiver also creates a set of (parallel) iOT-receiver messages with the bits of r and s as input. The receiver sends across to the sender the commitment c, the encryptions (e_0, e_1) and the iOT-receiver messages.

Upon receiving the receiver's first message, the sender in the semi-adaptive OT protocol uses its input strings m_0 and m_1 to create two circuits (this step is similar to the construction of iOT in Section 3). Based on the value of m_β, the garbled circuit GC_β is created.

- If $m_\beta = 1$ then GC_β is obliviously sampled such that it outputs $\perp$.
- Else, for $m_\beta = 0$ the garbled circuit GC_β is of the form (for $\beta \in \{0,1\}$): $\mathcal{C}[\beta, c, e_\beta](\cdot,\cdot)$, in the sense that each circuit is hardwired with a bit β, the receiver's commitment c and the receiver's commitment-encryption e_β; each circuit takes as input some randomness r and some randomness s and outputs 0 if all of the following conditions are satisfied: (a) c is a valid commitment to β under randomness r, (b) e_β is a valid encryption of r under randomness s. Otherwise it outputs $\perp$. The sender then garbles these circuits using the garbling algorithm (for $m_\beta = 0$) and oblivious garbling algorithm ($m_\beta = 1$) to create a pair of garbled circuits GC_0 and GC_1, along with the corresponding wire labels for their input bits.

The sender finally sends across GC_0 and GC_1 to the receiver. In parallel, the sender uses the iOT messages from the receiver to generate one iOT-sender message for each pair of wire labels, and also sends all of these to the receiver. The receiver uses the iOT-sender messages to recover the wire labels corresponding to its randomness strings (r, s) for both garbled circuits GC_0 and GC_1. It then evaluates GC_b on r and s by using the corresponding wire labels to recover the correct message m_b (it sets m_b to 0 if the GC_b evaluates to 0; otherwise, it sets m_b to 1).

Detailed Construction. Figure 5 presents a detailed description of our semi-adaptively simulation-secure two-message OT protocol π_{OT} in the CRS model from the following ingredients: (1) a two-message iOT protocol π_{iOT} with both receiver and sender oblivious sampleability in the CRS model and in the static corruption setting, (2) a trapdoor simulatable PKE, (3) an obliviously sampleable garbling scheme Garble, and (4) an equivocal commitment scheme Com (in the CRS model). We state the following theorem.

Theorem 6. *Assuming that: (1) π_{iOT} is a two-message iOT protocol with sender and receiver oblivious sampleability in the CRS model, (2) Com is an equivocal commitment scheme, (3) Garble is a private and an obliviously sampleable garbling scheme, and (4) the PKE scheme is trapdoor simulatable, π_{OT} is simulation-secure in the CRS-model against semi-adaptive malicious corruption of parties.*

The formal security proof is deferred to the full version of the paper. We present below an informal overview of the arguments for static and semi-adaptive security for our constructions.

Security against statically corrupt sender. The commitment c computationally hides the receiver's choice bit b because the encryption (e_0, e_1) computationally hides the receiver's randomness string r used for the commitment, and the iOT

Fig. 5. Semi-Adaptively Simulation-Secure Oblivious Transfer

π_{OT}

- **Public Inputs:** $\mathsf{crs}_{\mathsf{OT}} = (\mathsf{crs}_{\mathsf{iOT}}, \mathsf{crs}_{\mathsf{com}})$ where $\mathsf{crs}_{\mathsf{iOT}}$ and $\mathsf{crs}_{\mathsf{com}}$ are the setup strings of iOT and Com, respectively. Circuit $\mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, \mathsf{crs}_{\mathsf{pk}}, \beta, e, \mathsf{pk}](r, s) = 0$ if $c = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, b; r) \wedge e = \mathsf{Enc}(\mathsf{crs}_{\mathsf{pk}}, \mathsf{pk}, r; s)$; otherwise $\mathcal{C}$ outputs $\perp$.
- **Private Inputs:** S has input bits (m_0, m_1) where $m_0, m_1 \in \{0,1\}$; R has input choice bit b.
- **Primitives:** Let $\pi_{\mathsf{iOT}} = (\mathsf{iOTR}_1, \mathsf{iOTS}, \mathsf{iOTR}_2, \widetilde{\mathsf{iOTR}}, \widetilde{\mathsf{iOTS}})$ denote an iOT with (R, S)-oblivious sampling. Com is an equivocal commitment scheme. pk is the public key of a trapdoor simulatable PKE. $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \widetilde{\mathsf{Gb}})$ is an obliviously sampleable garbling scheme.

$\mathbf{OTR}_1(\mathsf{crs}_{\mathsf{OT}}, b)$:

- R commits to b using randomness r as $c = \mathsf{Com}(b; r)$.
- R encrypts r using randomness s as $e_b = \mathsf{Enc}(\mathsf{crs}_{\mathsf{pk}}, \mathsf{pk}, r; s)$. R samples $e_{\bar{b}} \leftarrow \mathsf{oEnc}(\mathsf{crs}_{\mathsf{pk}}, \mathsf{pk})$ obliviously. Let $t = (r||s)$ denote the commitment and encryption randomness, where $|t| = \ell$.
- R computes $\{\gamma_{0,i}, \gamma_{1,i}\}$ where $\gamma_{b,i} = \mathsf{iOTR}_1(\mathsf{crs}_{\mathsf{iOT}}, t_i)$ and $\gamma_{\bar{b},i} \leftarrow \widetilde{\mathsf{iOTR}}(\mathsf{crs}_{\mathsf{iOT}})$ for $i \in [\ell]$.
- R sends $\mathsf{M}_{\mathsf{R}} = \big(c, e_0, e_1, \{\gamma_{0,i}, \gamma_{1,i}\}_{i\in[\ell]}\big)$ as the receiver's OT message.

$\mathbf{OTS}(\mathsf{crs}_{\mathsf{OT}}, \mathsf{M}_{\mathsf{R}}, (m_0, m_1))$:
S runs the following algorithm for $\beta \in \{0,1\}$:

- If $m_\beta = 0$ then S computes the garbled circuit and input encoding as $\big(\mathrm{GC}_\beta, \{\mathsf{X}^0_{\beta,i}, \mathsf{X}^1_{\beta,i}\}_{i\in[\ell]}\big) = \mathsf{Gb}(1^\kappa, \mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, \mathsf{crs}_{\mathsf{pk}}, \beta, e_\beta, \mathsf{pk}])$. S computes the iOT sender messages as $\tau_{\beta,i} = \mathsf{iOTS}(\mathsf{crs}_{\mathsf{iOT}}, \gamma_{\beta,i}, (\mathsf{X}^0_{\beta,i}, \mathsf{X}^1_{\beta,i}))$ for $i \in [\ell]$.
- If $m_\beta = 1$ then S obliviously samples garbled circuit as $\mathrm{GC}_\beta = \widetilde{\mathsf{Gb}}(1^\kappa, \mathcal{C}[c, \mathsf{crs}_{\mathsf{com}}, \mathsf{crs}_{\mathsf{pk}}, \beta, e_\beta, \mathsf{pk}], \perp)$. S obliviously samples the iOT sender messages as $\tau_{\beta,i} \leftarrow \widetilde{\mathsf{iOTS}}(\mathsf{crs}_{\mathsf{iOT}}, \beta)$ for $i \in [\ell]$.

S sends $\mathsf{M}_{\mathsf{S}} = \{\mathrm{GC}_0, \{\tau_{0,i}\}_{i\in[\ell]}, \mathrm{GC}_1, \{\tau_{1,i}\}_{i\in[\ell]}\}$ as the sender's OT message

$\mathbf{OTR}_2(\mathsf{crs}_{\mathsf{OT}}, \mathsf{M}_{\mathsf{S}}, b)$:

- R computes the wire labels corresponding to commitment and encryption randomness t in GC_b as $\mathsf{Y}_i = \mathsf{iOTR}_2(\mathsf{crs}_{\mathsf{iOT}}, \tau_{b,i}, b)$ for $i \in [\ell]$.
- R outputs $m_b = 0$ if $\mathsf{Ev}(\mathrm{GC}, \{\mathsf{Y}_i\}_{i\in[\ell]}) = 0$ else R outputs $m_b = 1$.

messages sent by the receiver computationally hide the receiver's randomness string s for encryption.

A corrupt sender's messages can be extracted by a simulator $\mathcal{S}$. The simulator $\mathcal{S}$ constructs the commitment c in equivocal mode, i.e. $c = \mathsf{Com}(0; r_0) = \mathsf{Com}(1; r_1)$. The encryptions are set as follows - e_0 is an encryption of r_0 under randomness s_0 and e_1 is an encryption of r_1 under randomness s_1. $\mathcal{S}$ runs the two set of iOT messages correctly with input choice bits $t_0 = (r_0||s_0)$ and $t_1 = (r_1||s_1)$. Upon obtaining sender's OT message, the simulator decrypts input

wire labels for both GC_0 and GC_1. $\mathcal{S}$ evaluates GC_0 and GC_1 to extract m_0 and m_1 respectively.

Security against statically corrupt receiver. A corrupt receiver learns no information about $m_{\bar{b}}$. To see why this is the case, observe that the only information about $m_{\bar{b}}$ that the receiver could learn is from the garbled circuit $\mathrm{GC}_{\bar{b}}$. However, the receiver cannot evaluate $\mathrm{GC}_{\bar{b}}$ to anything other than $\perp$ since: (1) it cannot prove that c is a commitment to $\bar{b}$ under randomness r (this follows from the binding property of the commitment scheme), (2) it cannot prove that $e_{\bar{b}}$ decrypts to anything other than the commitment randomness r (this follows from the correctness of decryption for the PKE scheme), and (3) it cannot recover any input labels to $\mathrm{GC}_{\bar{b}}$ other than those corresponding to r (due to the sender privacy of the underlying iOT protocol). At this point, we invoke the privacy (when $m_{\bar{b}} = 0$) or oblivious sampleability (when $m_{\bar{b}} = 1$) of the garbling scheme to argue that the receiver learns no information about the message $m_{\bar{b}}$. Sender privacy follows from the privacy (when $m_{\bar{b}} = 0$) and oblivious sampleability (when $m_{\bar{b}} = 1$) of the garbling scheme, binding of the commitment scheme and the sender privacy of iOT. A corrupt receiver cannot obtain both wire labels for any input wire of a garbled circuit due to sender privacy of iOT. Given this argument holds, an honestly generated garbled circuit $\mathrm{GC}_{\bar{b}}$ is indistinguishable from an obliviously sampled one since in both cases the receiver evaluates $\mathrm{GC}_{\bar{b}}$ to $\perp$.

Next, we show a simulator that extracts a corrupt receiver's input. The receiver's input can be extracted using the secret key associated with the public key in the crs. The simulator decrypts e_0 and e_1 to obtain candidate randomness r_0 and r_1. It then checks whether $c = \mathsf{Com}(0; r_0)$ or $c = \mathsf{Com}(1; r_1)$. If both conditions are satisfied then the corrupt receiver has broken the binding property of the commitment scheme. Otherwise, the receiver's choice bit can be uniquely extracted. This completes our overview for static security.

Overview of Semi-adaptive Simulation-Security. Let us denote the set of OT messages for the bth branch (resp. $\bar{b}$th branch) as the bth set(resp. $\bar{b}$th set). Semi-adaptive simulation security considers two corruption scenarios: 1) the receiver gets corrupted post execution and the sender is statically corrupt, or 2) the receiver is statically corrupt and the sender gets corrupted post execution. In either of the cases, the simulator plays the role of the honest party which gets corrupted post-execution. The simulator needs to extract the input of the statically corrupt party. Also, when the honest party gets corrupted post execution, the simulator obtains the input of the honest party. The simulator needs to show randomness for the party such that the randomness is consistent with the party's input. We consider two corruption cases:

1. We first consider the case where the receiver gets corrupted post execution and the sender is statically corrupt. The simulator constructs the receiver OT message as described above. When the receiver gets corrupted post-execution the simulator shows randomness for the construction of e_b and claims that

$c = \mathsf{Com}(b; r_b)$. It also claims that $e_{\bar{b}}$ and the iOT sender messages for the $\bar{b}$th set were obliviously sampled. Indistinguishability follows due to the equivocal property of the commitment scheme, the oblivious ciphertext sampleability of the encryption scheme, and the receiver sampleability of iOT.

2. Next we consider the case where the sender gets corrupted post-execution and the receiver is statically corrupted. In this setting the simulator $\mathcal{S}$ extracts the choice bit b from the receiver's OT message. The simulator invokes the OT functionality $\mathcal{F}_{\mathsf{OT}}$ with b to obtain m_b. $\mathcal{S}$ constructs GC_b and the iOT sender messages for the bth set correctly. $\mathcal{S}$ also constructs $\mathrm{GC}_{\bar{b}}$ and iOT sender messages for the $\bar{b}$th set correctly as if $m_{\bar{b}} = 0$. This helps to equivocate the sender's view if $m_{\bar{b}}$ turns out to be 1 when the sender gets corrupted post-execution. We know that the evaluation of $\mathrm{GC}_{\bar{b}}$ always yields $\perp$ since c is not a valid commitment to $\bar{b}$. If the simulator is required to show randomness for $m_{\bar{b}} = 1$ then the simulator claims that $\mathrm{GC}_{\bar{b}}$ and the iOT sender messages for $\bar{b}$th set were obliviously sampled. This is indistinguishable from the real world execution where they were actually obliviously sampled. Thus, we rely on the sender oblivious sampling property of iOT and the oblivious sampling property of the garbling scheme to argue security.

5 Trapdoor Simulatable PKE from r-iOT

In this section, we show that any (two-message) $\mathsf{r\text{-}iOT}$ protocol implies a trapdoor simulatable PKE. The work of [CDMW09] constructed a two-round augmented NCE protocol from any trapdoor simulatable PKE scheme. This implies that any (two-message) $\mathsf{r\text{-}iOT}$ protocol implies a two-round augmented NCE protocol.

We actually show that any (two-message) iOT protocol satisfying *both* receiver *and* sender oblivious sampleability implies a trapdoor simulatable PKE. Since we already showed in Section 3 that any (two-message) $\mathsf{r\text{-}iOT}$ protocol implies that a (two-message) iOT protocol satisfying both receiver and sender oblivious sampleability, this yields our desired result.

Our Construction. Let $\mathsf{iOT} = (\mathsf{Setup}_{\mathsf{iOT}}, \mathsf{iOTR}_1, \mathsf{iOTS}, \mathsf{iOTR}_2)$ be an indistinguishability based OT. We construct a trapdoor simulatable PKE as follows:

- $\mathsf{Setup}(1^\kappa)$: Sample and output $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}_{\mathsf{iOT}}(1^\kappa)$.
- $\mathsf{Gen}(\mathsf{crs})$: Sample $\mathsf{M_R} = \mathsf{iOTR}_1(\mathsf{crs}, 0; \mathsf{rr_R})$ for uniformly sampled receiver randomness $\mathsf{rr_R}$. Output $(\mathsf{pk}, \mathsf{sk}) = (\mathsf{M_R}, \mathsf{rr_R})$.
- $\mathsf{Enc}(\mathsf{crs}, \mathsf{pk} = \mathsf{M_R}, m)$: Sample $m' \leftarrow \{0,1\}$ and generate the OT sender message $\mathsf{M_S} \leftarrow \mathsf{iOTS}(\mathsf{crs}, \mathsf{M_R}, (m, m'))$. Output the ciphertext $\mathsf{ct} = \mathsf{M_S}$.
- $\mathsf{Dec}(\mathsf{crs}, \mathsf{sk}, \mathsf{ct} = \mathsf{M_S})$: Output $m' = \mathsf{iOTR}_2(\mathsf{crs}, \mathsf{sk}, \mathsf{M_S})$.

Additionally, suppose that iOT is equipped with the oblivious sampling algorithms - $(\widetilde{\mathsf{iOTR}}, \widetilde{\mathsf{iOTS}})$ for the receiver and sender, and the corresponding inverting algorithms - $(\widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}, \widetilde{\mathsf{iOTS}_{\mathsf{Inv}}})$. We design the trapdoor simulatable PKE

to have oblivious sampling algorithms ($\mathsf{oGen}, \mathsf{oEnc}$) and randomness inverting algorithms ($\mathsf{rGen}, \mathsf{rEnc}$) defined as follows:

- $\mathsf{oGen}(\mathsf{crs}; \widetilde{\mathsf{rr}}_G)$: Sample $\widetilde{\mathsf{M}}_\mathsf{R} = \widetilde{\mathsf{iOTR}}(\mathsf{crs}; \widetilde{\mathsf{rr}}_G)$ and output $\widetilde{\mathsf{pk}} = \widetilde{\mathsf{M}}_\mathsf{R}$.
- $\mathsf{oEnc}(\mathsf{crs}; \widetilde{\mathsf{rr}}_E)$: Sample $m' \leftarrow \{0,1\}$ and $\widetilde{\mathsf{M}}_\mathsf{S} = \widetilde{\mathsf{iOTS}}(\mathsf{crs}, 0, m'; \widetilde{\mathsf{rr}}_E)$. Output $\widetilde{\mathsf{ct}} = \widetilde{\mathsf{M}}_\mathsf{S}$.
- $\mathsf{rGen}(\mathsf{crs}, \mathsf{rr}_G, \mathsf{td})$: Generate $\mathsf{M}_\mathsf{R} = \mathsf{iOTR}_1(\mathsf{crs}, 0; \mathsf{rr}_G)$ and output

$$\widehat{\mathsf{rr}}_G = \widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}(\mathsf{crs}, \mathsf{M}_\mathsf{R}, \mathsf{td}, \mathsf{rr}_G).$$

- $\mathsf{rEnc}(\mathsf{crs}, m, \mathsf{rr}_G, \mathsf{rr}_E, \mathsf{td})$: Generate the following:

$$\mathsf{M}_\mathsf{R} = \mathsf{iOTR}_1(\mathsf{crs}, 0; \mathsf{rr}_G), \quad \mathsf{M}_\mathsf{S} = \mathsf{iOTS}(\mathsf{crs}, \mathsf{M}_\mathsf{R}, (m, m); \mathsf{rr}_E),$$

  and output

$$\widehat{\mathsf{rr}}_E = \widetilde{\mathsf{iOTR}_{\mathsf{Inv}}}(\mathsf{crs}, \mathsf{M}_\mathsf{R}, \mathsf{M}_\mathsf{S}, \mathsf{td}, \mathsf{rr}_G, \mathsf{rr}_E).$$

Correctness of decryption follows immediately from the correctness of the underlying iOT scheme.

Theorem 7. *Our construction of trapdoor simulatable PKE is IND-CPA secure assuming that iOT satisfies indistinguishability security against a semi-honest sender and a semi-honest receiver.*

Theorem 8. *Our construction of trapdoor simulatable PKE satisfies trapdoor oblivious sampleability and randomness inversion assuming that iOT satisfies oblivious receiver and sender sampleability.*

The formal proofs are deferred to the full version of our paper. At a high level, ensuring oblivious sampleability (correspondingly randomness inversion) of the public key and ciphertexts in the resulting trapdoor simulatable PKE are relatively straightforward; one can simply reuse the receiver and sender oblivious sampling (correspondingly randomness inversion) algorithms provided by the iOT for obliviously sampling (correspondingly, inverting the randomness of) the public key and the ciphertext, respectively.

6 Instantiations of r-iOT from Concrete Assumptions

In this section we briefly discuss our instantiations of $\mathsf{r\text{-}iOT}$ from isogeny-based assumptions, CDH and LPN.

6.1 Instantiation from Isogeny-based Assumptions

In this section, we show how to construct a two-message r-iOT protocol secure against malicious adversaries in the CRS model from certain isogeny-based assumptions (notably, CSIDH [CLM+18] or CSI-FiSh [BKV19]). We base our construction on the existence of a secure *(restricted) effective group action* (EGA) equipped with appropriate computational hardness assumptions as described in [ADMP20]. We then rely on known instantiations of such a group action from the aforementioned isogeny-based assumptions.

In the rest of the section, we rely on the notations and formal definitions of EGA introduced in [ADMP20]. We refer the reader to [ADMP20] and to the full version of our paper for background material on group actions and EGA. We simply state here that our construction of r-iOT from group actions relies on the existence of a weak pseusorandom EGA, which is essentially the analogue of the DDH assumption in the context of group actions. As pointed out in [ADMP20], a weak pseudorandom EGA can be instantiated from isogeny-based assumptions, such as the decisional CSIDH assumption [CLM+18] and counterpart assumption in the setting of CSI-FiSh [BKV19].

The starting point of our construction of r-iOT is the construction of iOT from any (restricted) EGA proposed originally in [ADMP20]. This construction already satisfies indistinguishability-based security against maliciously corrupted sender and the receiver in the static corruption model. The key feature that this construction does not provide is receiver oblivious sampleability.

It turns out that we could argue that this construction satisfies receiver oblivious sampleability in a straightforward manner if we had the ability to sample obliviously from the "set" of a (restricted) EGA by "hashing into" the set. However, this is a well-known open problem in the isogeny literature and is likely to require fundamentally new ideas beyond state-of-the-art techniques for isogeny-based cryptography (see [Pet17, DMPS19, CPV20] for more details).

Our Construction. Our core technical centerpiece is a workaround for this wherein we settle for a weaker notion of *trapdoor oblivious sampleability* for the "set" of a (restricted) EGA. In other words, while it is hard to obliviously sample a "set" element in the plain model, one can obliviously sample a "set" element given a specially designed trapdoor (corresponding to some public CRS). This is the core idea behind our construction of r-iOT from (restricted) EGA. In view of the inherent restrictions outlined earlier, our workaround only allows us to achieve an r-iOT construction in the CRS model (and not in the plain model). Our construction of r-iOT from any weak pseudorandom (restricted) EGA is summarized in Fig. 6. Note that the sender and receiver algorithms remain unchanged from the original iOT construction due to [ADMP20].

Theorem 9. *Let $(G, X, \star)$ be a weak pseudorandom EGA (as introduced in [ADMP20]). The protocol in Figure 6 is an r-iOT protocol in the CRS model.*

The formal proof is deferred to the full version of our paper. We provide a proof overview here.

Fig. 6. Construction of r-iOT from weak pseudorandom EGA

Private Inputs: S has input bits (m_0, m_1); R has input choice bit b.
Primitives: $(G, X, \star)$ is a weak pseudorandom EGA with initial set element x_0 and $H : X^\ell \to \{0,1\}$ is a pairwise independent hash function.

Setup(1^κ):

- Sample and $g, h \leftarrow G$, and set $x_1 = g \star x_0$ and $x_2 = h \star x_0$.
- Sample $t \leftarrow G$ and compute $y_0 = t \star x_0, y_1 = t \star x_1$ and $y_2 = t \star x_2$.
- Set $\mathsf{crs}_{\mathsf{r\text{-}iOT}} = (x_0, x_1, x_2, y_0, y_1, y_2)$ and $\mathsf{td}_{\mathsf{r\text{-}iOT}} = (g, h, t)$.

r-iOTR$_1(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, b)$: Sample $s \leftarrow G$ and output $\mathsf{M_R}$ computed as follows:

$$\mathsf{M_R} = (u, v) = (s \star x_b, s \star y_b), \quad \mathsf{st} = s.$$

r-iOTS$(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \mathsf{M_R}, (m_0, m_1))$: For each $\beta \in \{0,1\}$, uniformly sample

$$\mathbf{r}_\beta = (r_{\beta,1}, \ldots, r_{\beta,\ell}) \leftarrow G^\ell, \quad \mathbf{b}_\beta = (b_{\beta,1}, \ldots, b_{\beta,\ell}) \leftarrow \{0,1\}^\ell.$$

For each $i \in [\ell]$ and for each $\beta \in \{0,1\}$, compute

$$c_{\beta,i} = \begin{cases} r_{\beta,i} \star x_\beta & \text{if } b_{\beta,i} = 0 \\ r_{\beta,i} \star y_\beta & \text{if } b_{\beta,i} = 1 \end{cases}, \quad c'_{\beta,i} = \begin{cases} r_{\beta,i} \star u & \text{if } b_{\beta,i} = 0 \\ r_{\beta,i} \star v & \text{if } b_{\beta,i} = 1. \end{cases}$$

For $\beta \in \{0,1\}$, define the vectors $\mathbf{c}_\beta, \mathbf{c}'_\beta \in X^\ell$ as:

$$\mathbf{c}_\beta := (c_{\beta,1}, \ldots, c_{\beta,\ell}), \quad \mathbf{c}'_\beta := (c'_{\beta,1}, \ldots, c'_{\beta,\ell}).$$

For $\beta \in \{0,1\}$, compute $z_\beta = H(\mathbf{c}'_\beta) \oplus m_\beta$, and output the sender message

$$\mathsf{M_S} = ((\mathbf{c}_0, z_0), (\mathbf{c}_1, z_1)).$$

r-iOTR$_2(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \mathsf{M_S}; \mathsf{st} = s)$: Output m_b computed as follows:

$$m_b = H(s \star c_1, \ldots, s \star c_\ell) \oplus z_b.$$

$\widetilde{\textbf{r-iOTR}}(\mathsf{crs}_{\mathsf{r\text{-}iOT}})$: Sample $s \leftarrow G$ and output $\widetilde{\mathsf{M_R}} = (s \star x_2, s \star y_2)$.

$\widetilde{\textbf{r-iOTR}}_{\textbf{Inv}}$ $(\mathsf{crs}_{\mathsf{r\text{-}iOT}}, \mathsf{M_R}, \mathsf{td}_{\mathsf{r\text{-}iOT}}, \mathsf{rr})$: Represent $\mathsf{rr} = (b||s)$ and $\mathsf{td}_{\mathsf{r\text{-}iOT}} = (g, h, t)$. If $b = 0$ then output $\hat{s} = sh^{-1}$, else output $\hat{s} = sgh^{-1}$.

Perfect Receiver Privacy. The receiver's choice bit b is perfectly hidden from the point of view of a (computationally unbounded) malicious receiver, even given $\mathsf{crs}_{\mathsf{r\text{-}iOT}}$ and $\mathsf{M_R} = (u, v)$. We show this by assuming $b = 1$ (the same argument holds when $b = 0$). If receiver computes (u, v) using randomness s when $b = 1$, then the same (u, v) can be shown as a valid receiver message for $b = 0$ using randomness $s' = sg$. In particular, we have $(u, v) = (sg \star x_0, sg \star y_0)$, since $x_1 = g \star x_0$ and $y_1 = t \star x_1 = tg \star x_0 = g \star (t \star x_0) = g \star y_0$.

Computational Sender Privacy. We show the following that there must be some bit $w \in \{0,1\}$ such that

$$\mathsf{r\text{-}iOTS}(\mathsf{crs}_{\mathsf{iOT}}, \mathsf{M}_{\mathsf{R}}, (m_0, m_1)) \overset{c}{\approx} \mathsf{r\text{-}iOTS}(\mathsf{crs}_{\mathsf{iOT}}, \mathsf{M}_{\mathsf{R}}, (m'_0, m'_1)),$$

where $m_{1-w} = m'_{1-w}$ and $m_w \neq m'_w$. We first modify the setup string to $\mathsf{crs}'_{\mathsf{r\text{-}iOT}}$ such that $y_0 = t_0 \star x_0$ and $y_1 = t_1 \star x_1$ where $t_0 \neq t_1$. We argue that $\mathsf{crs}_{\mathsf{r\text{-}iOT}}$ and $\mathsf{crs}'_{\mathsf{r\text{-}iOT}}$ are computationally indistinguishable based on the weak pseudorandomness of EGA.

Next, we argue that under the modified CRS $\mathsf{crs}'_{\mathsf{r\text{-}iOT}}$, there must be some bit $w \in \{0,1\}$ such that M_{S} statistically hides m_w irrespective of the manner in which a malicious receiver generates the message M_{R}. It allows us to move to a hybrid where the sender's message is modified to m'_w. The proof is very similar to the proof of Lemma 4.10 of [ADMP20].

Finally, we change the setup string back to $\mathsf{crs}_{\mathsf{r\text{-}iOT}}$ as in the real protocol. This switch is again computationally indistinguishable based on the weak pseudorandomness of EGA. At this point, the sender's message is distributed as $\mathsf{r\text{-}iOTS}(\mathsf{crs}_{\mathsf{iOT}}, \mathsf{M}_{\mathsf{R}}, (m'_0, m'_1))$, as desired. We refer to the full version of our paper for the formal proof.

Receiver Oblivious Sampleability. Finally, we claim that an obliviously sampleable receiver's message is distributed identically to an honestly generated message, even given the sampling randomness. In particular:

- If the receiver's choice bit $b = 0$, then $(u, v) = (s \star x_0, s \star y_0)$ generated using randomness s can be claimed as obliviously sampled using randomness $\hat{s} = sh^{-1}$, since $(u, v) = ((sh^{-1}) \star x_2, (sh^{-1}) \star y_2)$.
- If the receiver's choice bit $b = 1$, then $(u, v) = (s \star x_1, s \star y_1)$ generated using randomness s can be claimed as obliviously sampled using randomness $\hat{s} = sgh^{-1}$, since $(u, v) = ((sgh^{-1}) \star x_2, (sgh^{-1}) \star y_2)$.

6.2 Instantiation from CDH or LPN

To instantiate r-iOT from CDH or LPN, we rely on the iOT constructions of [DGH+20]. Specifically, Döttling *et al.* showed that iOT can be constructed from a weaker notion of OT called elementary OT, and they demonstrated instantiations of elementary OT from CDH or LPN assumption. The generic transformation of [DGH+20] is done in two steps: (1) they first show how to build iOT from an intermediate primitive called search OT via parallel repetition (which preserves receiver oblivious sampleability), (2) they show how to construct search OT from elementary OT where the receiver's message in search OT is identical to that of elementary OT.

Since the generic transformation of [DGH+20] does not affect *receiver* oblivious sampleability, it suffices to show that their elementary OT constructions from CDH or LPN *inherently* satisfy the receiver oblivious sampleability property. The corresponding proofs are immediate from the constructions in [DGH+20]. We refer the reader to the full version of our paper for more details.

Acknowledgements

The work of Pratik Sarkar is supported by the DARPA SIEVE project and NSF awards 1931714, 1414119.

References

AAAS+19. Gorjan Alagic, Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. *Status report on the first round of the NIST post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology . . . , 2019.

AASA+20. Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. Status report on the second round of the nist post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2020.

ADMP20. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *ASIACRYPT 2020, Part II*, LNCS, pages 411–439. Springer, Heidelberg, December 2020.

BBD+20. Zvika Brakerski, Pedro Branco, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Constant ciphertext-rate non-committing encryption from standard assumptions. In *TCC 2020, Part I*, LNCS, pages 58–87. Springer, Heidelberg, March 2020.

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. Springer, Heidelberg, November 2018.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *ASIACRYPT 2019, Part I*, LNCS, pages 227–247. Springer, Heidelberg, December 2019.

BL18. 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.

BLPV18. Fabrice Benhamouda, Huijia Lin, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Two-round adaptively secure multiparty computation from standard assumptions. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 175–205. Springer, Heidelberg, November 2018.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001.

CCD+20. Megan Chen, Ran Cohen, Jack Doerner, Yashvanth Kondi, Eysa Lee, Schuyler Rosefield, and Abhi Shelat. Multiparty generation of an RSA modulus. In *CRYPTO 2020*, pages 64–93, 2020.

CDD+04. Ran Canetti, Ivan Damgård, Stefan Dziembowski, Yuval Ishai, and Tal Malkin. Adaptive versus non-adaptive security of multi-party protocols. *Journal of Cryptology*, 17(3):153–207, June 2004.

CDMW09. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 287–302. Springer, Heidelberg, December 2009.

CFGN96. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.

CGP15. Ran Canetti, Shafi Goldwasser, and Oxana Poburinnaya. Adaptively secure two-party computation from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 557–585. Springer, Heidelberg, March 2015.

CJL+16. Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

CKWZ13. Seung Geol Choi, Jonathan Katz, Hoeteck Wee, and Hong-Sheng Zhou. Efficient, adaptively secure, and composable oblivious transfer with a single, global CRS. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 73–88. Springer, Heidelberg, February / March 2013.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

CLOS02. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002.

CPR17. Ran Canetti, Oxana Poburinnaya, and Mariana Raykova. Optimal-rate non-committing encryption. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 212–241. Springer, Heidelberg, December 2017.

CPV17a. Ran Canetti, Oxana Poburinnaya, and Muthuramakrishnan Venkitasubramaniam. Better two-round adaptive multi-party computation. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 396–427. Springer, Heidelberg, March 2017.

CPV17b. Ran Canetti, Oxana Poburinnaya, and Muthuramakrishnan Venkitasubramaniam. Equivocating yao: constant-round adaptively secure multiparty computation in the plain model. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 497–509. ACM Press, June 2017.

CPV20. Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. Rational isogenies from irrational endomorphisms. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 523–548. Springer, Heidelberg, May 2020.

CsW19. Ran Cohen, abhi shelat, and Daniel Wichs. Adaptively secure MPC with sublinear communication complexity. In Alexandra Boldyreva and Daniele

Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 30–60. Springer, Heidelberg, August 2019.

CSW20. Ran Canetti, Pratik Sarkar, and Xiao Wang. Efficient and round-optimal oblivious transfer and commitment with adaptive security. In *ASIACRYPT 2020*, pages 277–308, 2020.

DGH+20. Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 768–797. Springer, Heidelberg, May 2020.

DMPS19. Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In *ASIACRYPT 2019, Part I*, LNCS, pages 248–277. Springer, Heidelberg, December 2019.

EGL82. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 205–210. Plenum Press, New York, USA, 1982.

FMV19. Daniele Friolo, Daniel Masny, and Daniele Venturi. A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. In *TCC 2019, Part I*, LNCS, pages 111–130. Springer, Heidelberg, March 2019.

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.

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.

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.

GS12. Sanjam Garg and Amit Sahai. Adaptively secure multi-party computation with dishonest majority. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 105–123. Springer, Heidelberg, August 2012.

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.

GWZ09. Juan A. Garay, Daniel Wichs, and Hong-Sheng Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 505–523. Springer, Heidelberg, August 2009.

HK12. Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, January 2012.

HV15. Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On black-box complexity of universally composable security in the CRS model. In

Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 183–209. Springer, Heidelberg, November / December 2015.

IKN+17. Mihaela Ion, Ben Kreuter, Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, David Shanahan, and Moti Yung. Private intersection-sum protocol with applications to attributing aggregate ad conversions. Cryptology ePrint Archive, Report 2017/738, 2017. https://eprint.iacr.org/2017/738.

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.

LGdSG21. Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021*, pages 213–241, 2021.

LJA+18. Andrei Lapets, Frederick Jansen, Kinan Dak Albab, Rawane Issa, Lucy Qin, Mayank Varia, and Azer Bestavros. Accessible privacy-preserving web-based data analysis for assessing and addressing economic inequalities. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–5, 2018.

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.

MW16. Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 735–763. Springer, Heidelberg, May 2016.

Nao91. Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991.

NP01. Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457. ACM-SIAM, January 2001.

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 330–353. Springer, Heidelberg, December 2017.

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Heidelberg, August 2008.

Rab05. Michael O. Rabin. How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Report 2005/187, 2005. https://eprint.iacr.org/2005/187.

unb. Unbound security. https://www.unboundtech.com.

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.

YKT19. Yusuke Yoshida, Fuyuki Kitagawa, and Keisuke Tanaka. Non-committing encryption with quasi-optimal ciphertext-rate based on the DDH problem. In *ASIACRYPT 2019*, pages 128–158, 2019.