On the Round Complexity of Black-Box Secure MPC

Yuval Ishai[1], Dakshita Khurana[2], Amit Sahai[3], and Akshayaram Srinivasan[4]

[1] Technion
yuvali@cs.technion.ac.il
[2] UIUC
dakshita@illinois.edu
[3] UCLA
sahai@cs.ucla.edu
[4] Tata Institute of Fundamental Research
akshayaram.srinivasan@tifr.res.in

Abstract. We consider the question of minimizing the *round complexity* of secure multiparty computation (MPC) protocols that make a *black-box* use of simple cryptographic primitives with security against any number of malicious parties. In the plain model, previous black-box protocols required a high constant number of rounds (>15). This is far from the known lower bound of 4 rounds for protocols with black-box simulators. When allowing random oblivious transfer (OT) correlations, 2-round protocols making black-box use of a pseudorandom generator were known. However, such protocols were obtained via a round-collapsing "protocol garbling" technique that has poor concrete efficiency and makes non-black-box use of an underlying maliciously secure protocol.

We improve this state of affairs by presenting the following types of black-box protocols.

- **4-round "pairwise MPC" in the plain model.** This round-optimal protocol enables each ordered pair of parties to compute a function of both inputs whose output is delivered to the second party. The protocol makes black-box use of any public-key encryption (PKE) with pseudorandom public keys. As a special case, we get a black-box round-optimal realization of secure (copies of) OT between every ordered pair of parties.
- **2-round MPC from OT correlations.** This round-optimal protocol makes a black-box use of any *general* 2-round MPC protocol satisfying an augmented notion of *semi-honest* security. In the two-party case, this yields new kinds of 2-round black-box protocols.
- **5-round MPC in the plain model.** This protocol makes a black-box use of PKE with pseudorandom public keys, and 2-round oblivious transfer with "semi-malicious" security.

A key technical tool for the first result is a novel combination of split-state non-malleable codes (Dziembowski, Pietrzak, and Wichs, JACM '18) with standalone secure *two-party* protocols to construct *non-malleable two-party protocols*. The second result is based on a new round-optimized variant of the "IPS compiler" (Ishai, Prabhakaran and Sahai, Crypto '08). The third result is obtained via a specialized combination of these two techniques.

1 Introduction

Minimizing the *round complexity* of cryptographic protocols has been a central theme of research in the past few decades. Much of this research focused on the question of minimizing the round complexity of protocols for *secure multiparty computation* (MPC), both in the general case as well as for special tasks of interest such as zero-knowledge proofs, oblivious transfer (OT), or coin-tossing. This question is motivated not only by its direct relevance to the latency of protocols running over real-life networks, but also as an intriguing theoretical challenge that often inspires new ideas and serves as a test bed for new techniques.

The round complexity of MPC. We consider the standard setting of MPC with *an arbitrary number* of *malicious* parties, namely parties that are corrupted by a central adversary who may arbitrarily change their behavior. What do we know about the round complexity of MPC in this setting? Allowing a *common random string* (CRS) setup, it was recently shown [13, 28] that *2-round* MPC protocols are possible under the (minimal) assumption that 2-round OT exists in the CRS model. This round complexity is clearly optimal, even in the easier setting of *semi-honest* adversaries who send messages as instructed by the protocol. In the *plain model*, without any setup, a long line of works [27, 15, 6, 19, 54, 10, 9, 40] has culminated in *4-round* protocols that rely on the minimal assumption that a 4-round OT protocol exists [19]. This round complexity is known to be optimal for protocols that admit a *black-box simulator* [30, 52, 27]. All of the above 4-round protocols are of this kind.

Black-box constructions. Another central research theme in cryptography is obtaining *black-box constructions* of higher-level primitives from simpler lower-level primitives. A black-box construction of X from Y, also known as a (fully) black-box *reduction* from X to Y [59], specifies an implementation of X that only has oracle access to the input-output relation of Y, without being given any explicit representation of Y, e.g., in the form of a Boolean circuit or a Turing Machine. Moreover, it is required that the security reduction be black-box in the sense that any adversary A_X "attacking" X can be used as a black-box to obtain an adversary A_Y who obtains a similar advantage in attacking Y. Originating from the pioneering work of Impagliazzo and Rudich [43], a long line of works study the landscape of black-box reductions between natural cryptographic primitives. More relevant to our work is the effort to replace known instances of *non-black-box* constructions, where X requires access to the *code* of Y, by black-box constructions.

In the MPC context, early examples of results along this line include a black-box construction of constant-round *honest-majority* MPC protocols from one-way functions [22] (replacing an earlier non-black-box construction from [12]) and a black-box construction of malicious-secure OT from semi-honest OT [39] (replacing a non-black-box construction of [31]). Beyond the theoretical interest in understanding the tightness of the relation between primitives, the goal of replacing non-black-box constructions by black-box counterparts is strongly

motivated by asymptotic and concrete *efficiency*. A well-known example in the context of MPC is the non-black-box *OT extension* construction of Beaver [11], which was replaced by a much more efficient black-box construction from [44] that is commonly used as a basis for fast MPC implementations. We use the term *black-box MPC* to refer generically to an MPC protocol obtained via a black-box construction from simple low-level primitives (such as OT) that can be easily and efficiently constructed from standard cryptographic assumptions.

Round complexity of *black-box* MPC. Interestingly, all of the round-optimal MPC protocols in the standard setting we consider, including those mentioned above, make *non-black-box* use of the underlying primitives. In the case of 2-round MPC protocols in the CRS model, this is known to be inherent (even for the easier goal of semi-honest security), at least for black-box constructions from 2-round OT or any other *2-party* protocol [7]. However, no such impossibility result is known for 4-round MPC protocols in the *plain model*.

In the two-party case, a 4-round black-box protocol is known for *one-sided* functionalities that deliver output to only one of the two parties [57, 24]. The most general protocol of this kind makes a black-box use of any public-key encryption (PKE) with pseudorandom public keys, which can be easily constructed from most standard cryptographic assumptions [24]. This implies a similar 5-round protocol for two-sided functionalities.

In contrast, for a general number of parties, all known constant-round protocols are either complex and inefficient, or resort to idealized models such as the Random Oracle (RO) model to achieve better efficiency but only heuristic security. Despite the significant body of work on the round complexity of black-box MPC and related primitives in the plain model, the best exact round complexity that follows from existing works [49, 60, 32] is greater than 15 (see Section 1.2). Recent attempts to minimize round complexity [27, 15, 6, 19, 54, 10, 9] have led to complex protocols that make heavy non-black-box use of cryptography. This gap gives rise to the first motivating question for our work.

What is the minimal round complexity of black-box MPC in the plain model? Must we necessarily resort to idealized models to achieve simplicity and/or efficiency?

Round complexity of black-box protocol transformations. It turns out that if "plain model" is relaxed to allow a simple setup in the form of random *OT correlations* between each pair of parties, the first part of the above question has been settled. Concretely, given an OT correlation setup, which can be generated with good concrete efficiency [44, 14], there is a 2-round MPC protocol making a black-box use of a pseudorandom generator [26]. However, this 2-round protocol is quite complex and inefficient, as it is obtained by applying a heavily non-black-box "protocol garbling" transformation [28, 13] to an underlying multi-round (information-theoretic) MPC protocol. This not only hurts asymptotic and concrete efficiency, but also rules out applying this transformation while respecting a black-box use of an underlying primitive. The latter includes a

black-box use of an algebraic structure (e.g., a big finite field), a cryptographic primitive (e.g., homomorphic encryption or even a random oracle), or an ideal functionality oracle (e.g., OT or its arithmetic variant OLE). This is similar to the classical non-black-box protocol transformation from semi-honest MPC to malicious MPC, due to Goldreich, Micali, and Wigderson [31], which is limited in the same way.

In contrast, "black-box protocol transformations" from weak MPC protocols to stronger ones, commonly known as "MPC-in-the-head" transformations [46, 49, 50], have successfully avoided these limitations. In a nutshell, such transformations obtain a strong MPC protocol for f (say, with malicious security) by making a black-box use of any weak MPC protocol (say, with semi-honest security) for a *related* functionality f'. The relation between f and f' needs to be restricted in some way. Typically, f' is a next-message function of (an information-theoretic) weak MPC protocol for f.

This black-box protocol transformation paradigm, systematically studied in [48], has not only given rise to new theoretical feasibility and efficiency results, but it has also led to practical zero-knowledge proof systems [29, 5], digital signatures [16, 51], and MPC protocols [41]. The question we ask is whether one can obtain a similar black-box protocol transformation in the context of 2-round MPC with OT correlation setup:

> Are there useful kinds of "black-box protocol transformations" from 2-round semi-honest MPC to 2-round malicious MPC with OT correlation setup?

This question is particularly motivated in the two-party case, where there are many different techniques for efficient 2-round semi-honest protocols that make black-box use of algebraic or cryptographic primitives.

1.1 Our Contributions

We make progress on the aforementioned questions by obtaining the following types of round-efficient black-box protocols.

Black-box 4-Round "Pairwise MPC" in the Plain Model. Our first result addresses the first question by settling the round complexity of black-box MPC for a restricted but useful class of functionalities. Concretely, we get a 4-round black-box protocol for any *pairwise* MPC functionality that enable each ordered pair of parties to simultaneously compute a function of their inputs, whose output is delivered to the second party. The protocol makes a black-box use of any public-key encryption (PKE) with pseudorandom public keys (which can be instantiated based on CDH, LWE and LPN), similar to the 4-round 2-party OT protocol of [24].

Informal Theorem 1 *Let f be a pairwise MPC functionality. Assume the existence of a public-key encryption with pseudorandom public keys. There exists a four round black-box MPC protocol in the plain model that securely implements f against static corruptions of all-but-one parties.*

The central challenge in the pairwise MPC setting is to develop two-party protocols that remain secure *when executed in parallel*. We develop new black-box protocols for this setting, starting with the case of OT protocols, and generalizing via the result of [45] to any two-party functionality. To this end, a technical contribution of our work is a novel combination of split-state non-malleable codes [23, 18] with standalone secure *two-party* protocols to obtain black-box, *non-malleable* two-party protocols.

The resulting pairwise MPC can be used to generate OT correlations in a preprocessing phase, as required by the 2-round black-box protocol of [26]. This results in a 6-round MPC protocol making black-box use of PKE with pseudorandom public keys. While this already constitutes a major improvement over the state of the art, it is still two rounds away from the 4-round lower bound. Perhaps more importantly, as discussed above, the [26] approach employs a round-collapsing "protocol garbling" that limits its efficiency and applicability to protocols that make black-box use of algebraic or cryptographic primitives. Motivated by both limitations, we would like to replace the protocol garbling technique by a black-box protocol transformation that takes advantage of OT correlations.

An "IPS-style Compiler" for 2-round MPC. Our second main contribution is a new black-box protocol transformation obtained via a round-optimized variant of the "IPS compiler" [49]. This transformation uses a 2-round honest-majority MPC protocol from [47, 58] to transform in a black-box way any 2-round MPC protocol with an augmented variant of *semi-honest* security to obtain a 2-round MPC protocol with malicious security. The transformation relies on a special form of OT correlations (denoted as watchlist correlations) that can be generated via the above mentioned pairwise MPC functionality. Specifically, the watchlist correlations model outputs an n-party correlation between $(n-1)$ senders and a single receiver, where each sender S_i for $i \in [n-1]$ obtains a random set of m strings $x_{i,1}, \ldots, x_{i,m}$, and the receiver obtains a random subset $K \subset [m]$ of a fixed size, as well as the values $\{x_{i,j}\}_{i\in[n-1],j\in[k]}$. Combined with our first main result, this yields the same kind of 6-round black-box protocol obtained via [26], but with the advantage of making a black-box use of an augmented semi-honest protocol (as opposed to a non-black-box use of a malicious protocol incurred by the protocol garbling technique).

The augmented semi-honest security requirement combines the so-called *semi-malicious* security [8], which is satisfied by most natural 2-round semi-honest protocols, with a form of *adaptive security with erasures*. The latter is satisfied by all natural information-theoretic protocols (with standard forms of setup), as well as by computationally secure protocols with pre-processing. Concretely, we show the protocol from [26] in the OT correlations model and the protocol from [55] in the OLE correlations model satisfy augmented semi-honest security and thus, can be used in our compiler.

Informal Theorem 2 *Let f be an arbitrary multiparty functionality. Consider the client-server MPC protocol from [47] that securely computes f. Let f' be*

the function computed by the servers in this protocol. There exists a black-box transformation from a two-round MPC protocol for f' satisfying augmented semi-honest security to a two-round malicious secure protocol for computing f in the watchlist correlations model.

Towards concretely efficient 2-sided NISC. An interesting use case for the above result is the 2-round, secure *two-party* protocol in which *both* parties get an output. This should be contrasted with the standard notion of non-interactive secure computation (NISC) [45] that applies to one-sided functionalities. Note that this kind of *2-sided NISC* cannot be obtained by simply running two parallel instances of standard NISC, since even if we ignore parallel composition issues, there is no mechanism to enforce consistency between the inputs used in these instances (unless we rely on zero-knowledge proofs and make non-black-box use of cryptography). The only alternative black-box approach to 2-sided NISC over OT correlations we are aware of is via the protocol garbling technique that garbles the code of a malicious secure protocol and thus, has prohibitive computational and communication cost. Even in the 1-sided case, existing protocols from [45, 1, 56, 42, 17] are heavily tailored to specific garbling techniques and do not make a black-box use of an underlying semi-honest protocol.

We note that techniques developed in the context of an "IPS-style compiler" in the two-round setting gives a new approach for constructing protocols for the 2-sided NISC problem. Specifically, if we use [47, 58] as the outer protocol and use the simple two-sided version of Yao's protocol (using Boolean garbling in the OT correlations model) as the inner protocol, we obtain a 2-sided NISC protocol that is secure against malicious adversaries in the OT correlations model.[5] In Section 8.5 of the full version, we suggest some optimizations to improve the concrete efficiency.

Black-box 5-Round MPC in the Plain Model. Our third and final result uses a specialized combination of the previous contributions to get "one round away" from settling the main open question about the round complexity of black-box MPC. Concretely, we get a 5-round MPC protocol that makes a black-box use of PKE with pseudorandom public keys (as in the first contribution), along with any 2-round OT protocol with "semi-malicious" security. The latter security requirement is a very mild strengthening of semi-honest security in the context of 2-round OT protocols, and is satisfied by most 2-round OT protocols from the literature (for instance, it can be instantiated from standard assumptions such as DDH, LWE, QR).

Informal Theorem 3 *Let f be an arbitrary multiparty functionality. Assume the existence of a public key encryption with pseudorandom public keys and a two-round oblivious transfer protocol with semi-malicious security. There exists a five-round black-box protocol in the plain model that securely implements f against malicious adversaries that statically corrupts upto all-but-one parties.*

[5] As we noted before, for the case of constant number of parties, watchlist correlations reduces to standard OT correlations.

1.2 Related Work

In this subsection, we give a brief overview of the two main approaches taken by prior work obtaining black-box MPC protocols in the plain model.

Coin tossing based approach. The main idea in this approach is to use a black-box simulatable coin tossing protocol to setup a CRS and then use black-box MPC protocols (such as [GIS18]) in the CRS model. Roughly, to generate the CRS, the idea is for each party to commit to a random string r_i and in a later step, for all parties to reveal their coins. To ensure that malicious parties cannot set their randomness as a function of that of other honest players, players should use a (concurrent) non-malleable commitment in the commit phase.

But the main bottleneck to obtaining such a coin tossing protocol is achieving simulatability. To achieve the simulation guarantee and allow a simulator to "force" the output of the coin toss to be a certain value[6], one would need to rely zero-knowledge protocols, which if applied naively make non-black-box use of cryptography. Even if one were able to achieve simulation-based guarantees via a specific protocol, one would need to tailor this to prove statements about construction of bounded concurrent non-malleable commitment w.r.t. commitment against synchronising adversaries, for which no round efficient black-box constructions exist. More specifically, [35] gives a black-box protocol but the number of rounds of this protocol is greater than 18 (the coin tossing requires at least two more rounds. [36] gives a 3-round black-box construction of NMCom but is only secure in the standalone setting. The other round efficient constructions of concurrent NMCom [37, 20, 21, 53] make non-black use of cryptography.

IPS compiler based approach. The IPS compiler [49] gives a black-box MPC protocol in the OT hybrid model. The main challenge in instantiating this approach in the plain model is in constructing a protocol that securely realizes the ideal OT functionality. In particular, we need a protocol that realizes the ideal OT functionality between every ordered pair of parties. [60] gave a non-constant round black-box way to realize this which was improved by [32] who gave a constant round protocol. The main component in the constant round protocol is again a constant round black-box bounded concurrent non-malleable commitment wrt replacement (which is weaker than the traditional definition of non-malleable commitment wrt commitment). Even if we rely on a three-round black-box version of such a non-malleable commitment from [34], the OT protocol requires at least 12 rounds of communication. A straightforward way of combining this with the IPS approach incurs at least four more rounds.

2 Technical Overview

In this section, we provide an overview of the key technical ideas used in constructing a four round, black-box pairwise MPC in the plain model. One of the

[6] Note that this corresponds to the programmability requirement.

key building blocks used in this construction is a watchlist protocol. We give a construction of this protocol based on any public-key encryption with pseudorandom public keys and we elaborate on this next.

2.1 The Watchlist Protocol

We start by describing the ideal version of the watchlist functionality. The watchlist functionality is nothing but an implementation of a k-out-of-m oblivious transfer between each ordered pair of parties. Specifically, each ordered pair P_i and P_j execute a k-out-of-m OT where P_i acts as a receiver and P_j acts as a sender. We observe that the k-out-of-m OT is a one-sided functionality and hence, this can be realized if parties have pairwise access to independent copies of the ideal OT functionality [49, 45]. We call this as simultaneous secure OT and would like to securely realize this ideal functionality in the plain model in the presence of *arbitrary malicious* corruptions.

A Starting Point. A natural first attempt is to just have each pair of parties simultaneously execute a two-party secure protocol computing the k-out-of-m OT functionality. Such a protocol can be realized based on black-box use of any public key encryption scheme with pseudorandom public keys [24, 57].

Unfortunately, this *does not* securely emulate access to independent copies of the ideal OT functionality between pairs of participants, because this protocol satisfies only *stand-alone* security. It is easy to achieve OT that composes under parallel repetition with *fixed roles*, i.e., where many OT sessions are executed in parallel, and an adversary either corrupts multiple senders or multiple receivers but does *not* simultaneously corrupt (subsets of) senders and receivers. In particular, the stand-alone secure construction of OT from pseudorandom public keys in [24] already achieves this notion of parallel composition.

But in the (more general) simultaneous setting, an adversarial party P_i^* participates in many OT sessions simultaneously, as sender in some sessions and receiver in others. This gives P_i^* the opportunity to generate its own (e.g., sender) message in some OT session as a function of a message generated by an honest sender in a different OT session, thereby possibly making its own input depend on the input(s) of honest player(s). Clearly, this is disallowed by the ideal simultaneous OT functionality; but not prevented by standalone OT. Our first step towards addressing this vulnerability is to ensure that adversarial inputs are independent of the inputs of honest players.

As discussed in the introduction, we develop a novel approach to achieving such independence. In particular, we construct "non-malleable OT" that satisfies the following guarantees.

- **Receiver Security under Parallel Composition.** For every adversarial sender $\mathcal{A}^*$ that corrupts the OT sender (or resp., multiple senders in any parallel composition of the OT protocol), there exists a simulator that simulates the view of $\mathcal{A}^*$ with black-box access to (resp., copies of) the ideal OT functionality. This follows automatically from simulation-based security

against malicious senders (resp., in the parallel composition setting) of the underlying two-party secure protocol $\Pi_{\mathcal{F}}$.

- **Non-Malleability.** Informally, here we consider a man-in-the-middle adversary MIM that acts as a receiver in a subset of OT sessions (that we refer to as "left" sessions) and as sender in a different subset of OT sessions (that we refer to as "right" sessions).
  We require the existence of a *simulator-extractor* $\mathsf{Sim\text{-}Ext}$, that given the inputs of all honest receivers (participating in all right sessions), is able to extract all the implicit inputs used by the MIM *in all its right sessions.* Crucially, $\mathsf{Sim\text{-}Ext}$ *does not* have access to the inputs of honest senders (participating in the left sessions).
  This is the key property that prevents an adversarial sender from "copying" the inputs of honest senders, or more generally, generating its inputs as a function of honest senders' inputs. Achieving this property will be a key technical focus of our work.

In what follows, we provide an overview of our construction of non-malleable OT. Then, in Section 2.1, we discuss why any non-malleable OT protocol satisfying these properties almost directly implies pairwise ideal OT functionality (or, simulataneous secure OT), and therefore also securely realizes watchlists.

Towards Non-Malleable OT We take inspiration from recent works that use non-malleable codes (introduced in [23]) to build cryptographic primitives like non-malleable commitments [36], and non-malleable multi-prover interactive proofs [33].

In more detail, we will build non-malleable OT by combining *parallel composable* two-party secure computation with (an) appropriate (variant of) split-state non-malleable codes. Such codes are specified by encoding and decoding algorithms $(\mathsf{Enc}, \mathsf{Dec})$. The encoding algorithm Enc is a randomized algorithm that encodes any message m into a codeword consisting of two parts or "states" (L, R), and the decoding algorithm Dec on input a codeword returns the underlying message. The security property is that for every pair of tampering functions (f, g) with no fixed points, the (distribution of) $\widetilde{m} \leftarrow \mathsf{Dec}(f(\mathsf{L}), g(\mathsf{R}))$, where $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$, is independent of m. We now describe (a simplified variant of) our construction.

Our Construction For simplicity, we will focus on the special case of implementing non-malleable 1-*out-of*-2 *OT*, but our techniques immediately extend to the more general setting of k-out-of-m OT. To prevent obvious copying attacks, we will assign to each party a unique tag or identity.

Our construction of non-malleable OT simply involves *executing a secure two-party protocol* Π between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$, for a special functionality $\mathcal{F}$. Before describing this functionality, we discuss the inputs of participants to this functionality.

The sender $\mathcal{S}$ with on input $(\mathsf{m}_0, \mathsf{m}_1)$ and tag encodes these messages using an appropriate split-state non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$. Specifically, $\mathcal{S}$ computes

$\mathsf{L}_0, \mathsf{R}_0 \leftarrow \mathsf{Enc}(\mathsf{m}_0||\mathsf{tag})$ and $\mathsf{L}_1, \mathsf{R}_1 \leftarrow \mathsf{Enc}(\mathsf{m}_1||\mathsf{tag})$. The receiver $\mathcal{R}$ obtains as input a choice bit $\mathsf{b} \in \{0,1\}$, and samples uniformly random $c \in \{0,1\}$. $\mathcal{S}$ and $\mathcal{R}$ then invoke a two-party secure protocol $\Pi_{\mathcal{F}}$ to compute functionality $\mathcal{F}$ described in Figure 1. In addition, $\mathcal{S}$ sends tag to $\mathcal{R}$.

Sender Inputs: $\mathsf{m}_0, \mathsf{L}_0, \mathsf{R}_0, \mathsf{m}_1, \mathsf{L}_1, \mathsf{R}_1, \mathsf{tag}$. **Receiver Inputs:** b, c.

The functionality $\mathcal{F}(\mathsf{m}_0, \mathsf{L}_0, \mathsf{R}_0, \mathsf{m}_1, \mathsf{L}_1, \mathsf{R}_1, \mathsf{b}, c, \mathsf{tag})$ is defined as follows.

1. If $\mathsf{Dec}(\mathsf{L}_0, \mathsf{R}_0) \neq (\mathsf{m}_0||\mathsf{tag})$ or $\mathsf{Dec}(\mathsf{L}_1, \mathsf{R}_1) \neq (\mathsf{m}_1||\mathsf{tag})$, output $\perp$.
2. If $c = 0$, output $(\mathsf{m}_b, \mathsf{L}_0, \mathsf{L}_1)$ and output $(\mathsf{m}_b, \mathsf{R}_0, \mathsf{R}_1)$, otherwise.

Fig. 1: The functionality $\mathcal{F}$

We note that the ideal functionality $\mathcal{F}$ reveals m_b to $\mathcal{R}$, and in addition, reveals either *only* $(\mathsf{L}_0, \mathsf{L}_1)$ or *only* $(\mathsf{R}_0, \mathsf{R}_1)$. Because any split-state non-malleable code is also a 2-out-of-2 secret sharing scheme [4], the shares L_{1-b} and R_{1-b} each statistically hide m_{1-b} from $\mathcal{R}$. It is also clear that protocol Π makes only black-box use of the underlying two-party computation protocol.

We show that (an appropriate k-out-of-m variant of) the protocol sketched above securely realizes non-malleable OT, even when Π itself is only parallel composable secure (but may be completely malleable).

Proving Sender Non-Malleability. For ease of exposition, let's consider a simpler man-in-the-middle adversary (MIM) that participates in a single left session as receiver, and a single right session as sender. We will also assume that the MIM never sends messages that cause an honest party to abort. Additionally, the underlying secure two-party protocol Π will be a round optimal (four round) protocol with sequential messages, and has the following specific structure. Namely, it will require the receiver to commit to its input b in the first round of the protocol, and at the same time, it will be *delayed-input* w.r.t. receiver input c, which will be chosen by the receiver immediately before the third round begins. Finally, it will require the inputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{L}_0, \mathsf{R}_0, \mathsf{L}_1, \mathsf{R}_1, \mathsf{tag})$ of the sender to be committed in the second round of the protocol, *before c is chosen by the receiver.*

First Attempt: An Alternate Extraction Mechanism. One possible way to extract sender inputs from the right execution, is to execute the simulator of the underlying two-party protocol Π. Unfortunately, this fails because Π is only parallel composable secure, and extracting from the right execution automatically reveals honest sender inputs from the left execution.

Instead, we will use the specific way that sender inputs are encoded to introduce an *alternate* extraction mechanism. Specifically, one could imagine rewinding the third and the fourth round message of Π, using inputs $c = 0$ and $c = 1$ on behalf of the honest receiver in the real and rewinding threads, respectively.

By our assumption, the adversary is non-aborting. Therefore, we expect to obtain outputs $(\widetilde{\mathsf{L}}_0, \widetilde{\mathsf{L}}_1)$ and $(\widetilde{\mathsf{R}}_0, \widetilde{\mathsf{R}}_1)$ in the right session in the real and rewinding threads respectively. At this point, we can use the decoder of the non-malleable code to obtain $(\widetilde{\mathsf{m}}_0, \widetilde{\mathsf{m}}_1)$, which, by correctness of the two-party protocol, should correspond to the implicit inputs of the MIMin the right session.

It doesn't seem like this argument gives us much (yet): rewinding the MIM's third and fourth rounds would also end up rewinding the third and the fourth rounds of the left execution with (possibly different) inputs $\widetilde{c}, \widehat{c}$ used by the MIMin the main and rewinding threads. Thus, it may seem like we are back to square one: it may not be possible to hide the inputs of the honest sender in the presence of such rewinding.

Towards Resolving the Extraction Bottleneck: 1*-Rewind Sender Security.* To tackle this problem, our first step will be to require that Π satisfy a stronger security property: 1*-rewind sender security*. Roughly, this means that any adversarial receiver $\mathcal{R}^*$ that rewinds the honest sender one time in the third and fourth rounds, using (possibly different) inputs $\widetilde{c}, \widehat{c}$ in the main and rewinding threads, does not recover any information beyond the output of $\mathcal{F}$ on inputs $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{L}_0, \mathsf{L}_1, \mathsf{R}_0, \mathsf{R}_1, \widetilde{\mathsf{b}}, \widetilde{c})$ and $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{L}_0, \mathsf{L}_1, \mathsf{R}_0, \mathsf{R}_1, \widetilde{\mathsf{b}}, \widehat{c})$. We formalize this by requiring the existence of a specific type of simulator: this simulator generates a view for $\mathcal{R}^*$ in the main thread given only $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{L}_0, \mathsf{L}_1)$ and a view for $\mathcal{R}^*$ in the rewinding thread given only $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{R}_0, \mathsf{R}_1)$ (or vice-versa). Now, it may seem like this type of simulator may not be very meaningful, since the sum total of this information could essentially allow the receiver to recover $\mathsf{m}_{1-\widetilde{\mathsf{b}}}$ by combining $\mathsf{L}_{1-\widetilde{\mathsf{b}}}$ with $\mathsf{R}_{1-\widetilde{\mathsf{b}}}$.

However, the fact that $(\mathsf{L}_0, \mathsf{L}_1)$ and $(\mathsf{R}_0, \mathsf{R}_1)$ are made available in *separate* threads can be exploited argue that the MIM's input must be independent of $\mathsf{m}_{1-\widetilde{\mathsf{b}}}$, as we discuss next.

Alternative Simulation. Let us go back to our alternate extraction mechanism discussed earlier, where w.l.o.g. the third and fourth round messages of Π are rewound with (honest) receiver input set to $c = 0$ in the main and $c = 1$ in the rewinding thread, respectively. This means that in the main thread, the challenger obtains output $(\widetilde{\mathsf{L}}_0, \widetilde{\mathsf{L}}_1)$ in the right session. In the rewind thread, setting $c = 1$, the challenger obtains outputs $(\widetilde{\mathsf{R}}_0, \widetilde{\mathsf{R}}_1)$. Meanwhile the real and rewinding left executions will simulated using only $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{L}_0, \mathsf{L}_1)$ and $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{R}_0, \mathsf{R}_1)$ (or vice-versa) respectively, as described above. *This means that in the main thread, the* MIM *outputs* $(\widetilde{\mathsf{L}}_0, \widetilde{\mathsf{L}}_1)$ *as a function of only* $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{L}_0, \mathsf{L}_1)$*, and in the rewinding thread, the* MIM *outputs* $(\widetilde{\mathsf{R}}_0, \widetilde{\mathsf{R}}_1)$ *as a function of only* $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{R}_0, \mathsf{R}_1)$.[7]

We formalize this intuition to argue that the MIM's behaviour naturally gives rise to a split-state tampering function family. Here, one tampering function

[7] Actually, the MIMmay also output $(\widetilde{\mathsf{L}}_0, \widetilde{\mathsf{L}}_1)$ as a function of only $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{R}_0, \mathsf{R}_1)$, and $(\widetilde{\mathsf{R}}_0, \widetilde{\mathsf{R}}_1)$ as a function of only $(\mathsf{m}_{\widetilde{\mathsf{b}}}, \mathsf{L}_0, \mathsf{L}_1)$. We use codes satisfying an additional *symmetric decoding* property to account for this case.

corresponds to the MIM's functionality in the main thread, and the other corresponds to the MIM's functionality in the rewinding thread. This allows us to rely on the non-malleability of the underlying encoding scheme to switch from generating $\mathsf{L}_{1-\tilde{\mathsf{b}}}, \mathsf{R}_{1-\tilde{\mathsf{b}}}$ as an encoding of $m_{1-\mathsf{b}}$, to generating it as an encoding of a dummy value.

This completes a simplified description of the main ideas in our protocol. We swept several details under the rug but point out one important detail below.

Many-many Non-malleability. Recall that we simplified things earlier, to focus on a setting where the MIM participates in a single left session as receiver and a single right session as sender. For our application to watchlists, we require security against adversaries that participate in *multiple* left and right sessions.

To achieve security in this setting, we will rely on *many-many* non-malleable codes (that are implied by one-many non-malleable codes [18]) that achieve security in the presence of multiple tamperings of a single codeword [18]. Moreover, in order to deal with adversaries that may abort arbitrarily, we will modify the functionality $\mathcal{F}$. Instead of encoding (m_0, m_1) a single time, the sender generates λ (where λ is the security parameter) fresh encodings $\{(\mathsf{L}^i_{\mathsf{b}}, \mathsf{R}^i_{\mathsf{b}})\}_{i\in[\lambda],\mathsf{b}\in\{0,1\}}$ of m_0 and m_1. The receiver picks λ choice bits $c_1, \ldots, c_\lambda$ instead of a single bit c. The functionality $\mathcal{F}$ checks if for every $i \in [\lambda], \mathsf{b} \in \{0,1\}$, $\{(\mathsf{L}^i_{\mathsf{b}}, \mathsf{R}^i_{\mathsf{b}})\}_{i\in[\lambda],\mathsf{b}\in\{0,1\}}$ encode m_{b}. If the check fails, $\mathcal{F}$ outputs $\perp$. If it passes, then for every $i \in [\lambda]$, it outputs $(\mathsf{L}^i_0, \mathsf{L}^i_1)$ if $c_i = 0$ and otherwise, outputs $(\mathsf{R}^i_0, \mathsf{R}^i_1)$.

This helps ensure that for every adversary MIM that completes a main thread (without aborting) given honest receiver input $c = c_1, \ldots, c_\lambda$, there is (w.h.p.) a rewinding thread with a different choice $c' = c'_1, \ldots, c'_\lambda$ of honest receiver input, that is also completed by the MIM. We then rely on any index i for which $c_i \neq c'_i$ to carry out the argument described above. Additional details of our non-malleable OT protocol can be found in Section 5.1 in the full version.

From Non-Malleable OT to Watchlists We note that that our OT protocol, as described above, prohibits an adversarial sender from generating its generating its inputs as a function of honest senders' inputs.

One could ask for an even stronger property, requiring the inputs of adversarial receivers to be independent of the honest receivers' inputs. At first glance, this stronger property appears to be necessary, since pairwise access to ideal OTs would actually enforce that all adversarial receiver inputs are independent of the inputs of honest receivers.

But upon taking a closer look, we realize that non-malleable OT as described in the previous section actually suffices to construct watchlists with security in the real/ideal paradigm. Intuitively, this is because the outputs of honest parties are affected only by the inputs of the adversarial senders, and are unaffected by the inputs of adversarial receivers. In other words, even if adversarial receivers manage to have their inputs depend on the inputs of the honest receivers, this cannot affect the joint distribution of their view and the outputs of honest parties in the ideal world. We formalize this intuition and show that non-malleable OT generically implies a protocol for securely realizing the watchlist functionality.

The only missing ingredient in our description is the 1-rewind sender secure protocol, which we describe next.

Constructing a 1-Rewind Sender Secure Protocol In our actual construction of non-malleable OT, the receiver inputs $(c_1, \ldots c_\lambda)$ do not need to remain hidden from a corrupted sender. In particular, all we need is for the protocol to allow for delayed *function* selection, where the function to be computed (defined by $c_1, \ldots, c_\lambda$) is selected by the receiver in the third round. Given this, the 1-rewinding security property translates to requiring that any corrupt receiver which rewinds the third and the fourth round messages of the sender by providing (possibly) different functions learns nothing beyond the output of these two functions on sender and receiver inputs that were fixed in the first two rounds.

We will design such a 2-party protocol for $\mathsf{NC1}$ circuits[8] by relying on a different variant [46, 49, 45] of the IPS paradigm. Specifically, we will use the same 2-client m-server outer protocol [58] that was discussed at the beginning of the overview, and combine it an inner protocol that is based a variant of Yao's garbled circuits [61]. Yao's protocol also allows for the garbled circuits to be generated in the final round, which immediately gives us the delayed function selection property. Importantly, since we only care about parallel composable security in the resulting two-party protocol, parallel composable but possibly malleable 1-rewind secure OT will suffice to implement watchlists in this setting. We slightly generalize the works of [57, 24] to obtain a maliciously secure OT that satisfies 1-rewind sender security and makes black-box use of a PKE with pseudorandom public keys. We refer the reader to Appendix C of the full version for the details of constructing the secure computation protocol and to Appendix D of the full version for the construction of a 1-rewind sender secure OT protocol.

Immediate Application: Black-box Simultaneous Two-Party Computation Plugging the resulting simultaneous OT protocol in place of ideal OT, into the non-interactive two-party secure black-box computation protocol of [45], yields a round optimal two-party *simultaneous* secure computation, from black-box use of any PKE with pseudorandom public keys.

Organization. Due to lack of space, we include our construction of non-malleable OT in the body of the paper, and defer remaining protocols to the full version.

3 Preliminaries and Definitions

Let λ denote the security parameter. A function $\mu(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is said to be negligible if for any polynomial $\mathsf{poly}(\cdot)$ there exists λ_0 such that for all $\lambda > \lambda_0$ we have $\mu(\lambda) < \frac{1}{\mathsf{poly}(\lambda)}$. We will use $\mathsf{negl}(\cdot)$ to denote an unspecified negligible

[8] We show in Section 5.1 of the full version that 1-rewind secure 2PC for $\mathsf{NC1}$ circuits suffices to obtain non-malleable OT.

function and $\mathsf{poly}(\cdot)$ to denote an unspecified polynomial function. We use Δ to denote the statistical distance.

For a probabilistic algorithm A, we denote $A(x;r)$ to be the output of A on input x with the content of the random tape being r. When r is omitted, $A(x)$ denotes a distribution. For a finite set S, we denote $x \leftarrow S$ as the process of sampling x uniformly from the set S. We will use PPT to denote Probabilistic Polynomial Time algorithm.

3.1 Non-malleable Codes

We will use non-malleable codes in the split-state model, that are one-many secure and satisfy a special augmented non-malleability [2] property, as discussed below.

Definition 1 (One-many augmented split-state non-malleable codes). *Fix any polynomials $\ell(\cdot), p(\cdot)$. An $\ell(\cdot)$-augmented non-malleable code with error $\epsilon(\cdot)$ for messages $m \in \{0,1\}^{p(\lambda)}$ consists of algorithms $\mathsf{NM.Code}, \mathsf{NM.Decode}$ where $\mathsf{NM.Code}(m) \to (L, R)$ such that for every $m \in \{0,1\}^{p(\lambda)}$,*

$$\mathsf{NM.Decode}(\mathsf{NM.Code}(m)) = m$$

and for every set of functions $f = (f_1, f_2, \ldots f_{\ell(\lambda)}), g = (g_1, g_2, \ldots g_{\ell(\lambda)})$ there exists a random variable $\mathcal{D}_{f,g}$ on $\{\{0,1\}^{p(\lambda)} \cup \mathsf{same}^*\}^{\ell(\lambda)}$ which is independent of the randomness in $\mathsf{NM.Code}$ such that for all messages $m \in \{0,1\}^{p(\lambda)}$ it holds that

$$\Delta\left(\left(\mathsf{R}, \{\mathsf{NM.Decode}\big(f_i(\mathsf{L}), g_i(\mathsf{R})\}_{i\in[\ell(\lambda)]}\right)\right), (\mathsf{replace}(\mathcal{D}_{f,g}, m))\right) \leq \epsilon(\lambda)$$

$$\Delta\left(\left(\mathsf{R}, \{\mathsf{NM.Decode}\big(g_i(\mathsf{R}), f_i(\mathsf{L})\}_{i\in[\ell(\lambda)]}\right)\right), (\mathsf{replace}(\mathcal{D}_{f,g}, m))\right) \leq \epsilon(\lambda)$$

where $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{NM.Code}(\mathsf{m})$ and the function $\mathsf{replace} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ replaces all occurrences of same^* in its first input with its second input s, and outputs the result.

It was shown in [34, 38, 3] that the CGL one-many non-malleable codes constructed in [18] are also one-many *augmented* non-malleable codes. But we point out that in this definition, we also require messages obtained by decoding the tampered codewords with *left and right shares interchanged* to be unrelated with the original message. It is easy to see that this property is satisfied by any non-malleable code with symmetric decoding (i.e. where $\mathsf{NMDec}(\mathsf{L}, \mathsf{R}) = \mathsf{NMDec}(\mathsf{R}, \mathsf{L})$). This property can be achieved, as observed in [33], by modifying any split-state code to attach a special symbol "ℓ" to the left part of the codeword, and a special symbol "r" to the right part of the codeword. This yields the following imported theorem:

Theorem 1. *(Imported.) [33, 34] For every polynomial $\ell(\cdot)$, there exists a polynomial $q(\cdot)$ such that for every $\lambda \in \mathbb{N}$, there exists an explicit ℓ-augmented, split-state non-malleable code satisfying Definition 1 with efficient encoding and decoding algorithms with code length $q(\lambda)$, rate $q(\lambda)^{-\Omega(1)}$ and error $2^{-q(\lambda)^{\Omega(1)}}$.*

3.2 Low-Depth Proofs

We will describe how any computation that can be verified by a family of polynomial sized ciruits can be transformed into a proof that is verifiable by a family of circuits in $\mathsf{NC1}$. Let R be an efficiently computable binary relation. Let L be the language consisting of statements in R, i.e. for which $R(x) = 1$.

Definition 2 (Low-Depth Non-Interactive Proofs). *A low-depth non-interactive proof with perfect completeness and soundness for a relation R consists of an (efficient) prover P and a verifier V that satisfy:*

- **Perfect completeness.** *A proof system is perfectly complete if an honest prover can always convince an honest verifier. For all $x \in L$ we have*

$$\Pr[V(\pi) = 1 | \pi \leftarrow P(x)] = 1$$

- **Perfect soundness.** *A proof system is perfectly sound if it is infeasible to convince an honest verifier when the statement is false. For all $x \notin L$ and all (even unbounded) adversaries $\mathcal{A}$ we have*

$$Pr[V(x, \pi) = 1 | \pi \leftarrow \mathcal{A}(x)] = 0.$$

- **Low depth.** *The verifier V can be implemented in* $\mathsf{NC1}$.

We outline a simple construction of a low-depth non-interactive proof, borrowed from [25]. The prover P executes the verification circuit on x and generates the proof as the sequential concatenation (in some specified order) of the bit values assigned to the individual wires of the circuit computing R. The verifier V proceeds by checking consistency of the values assigned to the internal wires of the circuit for each gate. In particular for each gate in the verification circuit the verifier checks if the wire vales provided in the proof represent a correct evaluation of the gate. Since the verification corresponding to each gate can be done independent of every other gate and in constant depth, we have that V itself is constant depth.

Looking ahead, our construction of non-malleable OT makes use of a (malleable) two-party computation protocol for $\mathsf{NC1}$ that must verify validity of a non-malleable code. We rely on low-depth proofs to ensure that the two-party computation protocol only performs $\mathsf{NC1}$ computations.

3.3 1-Rewind Sender-Secure Two-Party Computation

Let us consider a protocol Π between two parties, namely, the sender $\mathcal{S}$ and the receiver $\mathcal{R}$. The sender holds a private input $x_{\mathcal{S}}$ and the receiver holds a private input $x_{\mathcal{R}}$ and they wish to compute some function of their private inputs securely with the receiver obtains the output of the function. We want this protocol to satisfy:

– (Delayed-function selection) The function to be securely computed is only decided in the third round by the receiver $\mathcal{R}$. That is, the third round message contains the explicit description of the function f to be computed and the first two messages depend only on the size of the function.
– (1-Rewinding Security) Any malicious receiver that rewinds the third and fourth rounds of the protocol once (by possibly giving different functions f_0, f_1) cannot learn anything about the sender's inputs except the output on these two functions.

The syntax of the protocol and the two properties are formalized below.

Syntax. The special two party protocol Π is given by a tuple of algorithms $(\Pi_1, \Pi_2, \Pi_3, \Pi_4, \mathsf{out}_\Pi)$. Π_1 and Π_3 are the next message functions run by the receiver $\mathcal{R}$ and Π_2 and Π_4 are the next message functions run by the sender $\mathcal{S}$. At the end of the protocol, $\mathcal{R}$ runs out_Π on the transcript, its input and the random tape to get the output of the protocol. We use π_r to denote the message sent in the protocol Π in round r for every $r \in [4]$.

Definition 3. *Let $\Pi = (\Pi_1, \Pi_2, \Pi_3, \Pi_4, \mathsf{out}_\Pi)$ be a 4-round protocol between a receiver $\mathcal{R}$ and a sender $\mathcal{S}$ with the receiver computing the output at the end of the fourth round. We say that Π is a 1-rewinding sender secure protocol with delayed function selection for ^{1} circuits if it satisfies:*

– ***Delayed Function Selection.*** *The first and second message functions Π_1, Π_2 take as input the size of the function $f \in^1$ to be securely computed and are otherwise, independent of the function description. The third round message from $\mathcal{R}$ contains the explicit description of the function f to be computed.*
– ***Receiver Security.*** *For every malicious PPT adversary $\mathcal{A}$ that corrupts the sender, there exists an expected polynomial (black-box) simulator $\mathsf{Sim}_\mathcal{R} = (\mathsf{Sim}^1_\mathcal{R}, \mathsf{Sim}^2_\mathcal{R})$ such that for all choices of honest receiver input $x_\mathcal{R}$ and the function $f \in^1$, the joint distribution of the view of $\mathcal{A}$ and $\mathcal{R}$'s output in the real execution is computationally indistinguishable to the output of the ideal experiment described in Figure 2.*
– ***1-Rewinding Sender Security.*** *For every malicious adversary $\mathcal{A}$, corrupting the receiver, there exists an expected polynomial time simulators $\mathsf{Sim}_\mathcal{S} = (\mathsf{Sim}^1_\mathcal{S}, \mathsf{Sim}^2_\mathcal{S})$ such that for every choice of sender's input $x_\mathcal{S}$, we have:*

$$\mathsf{Expt}_1(\mathcal{A}, \Pi, x_\mathcal{R}, x_\mathcal{S}) \approx_c \mathsf{Expt}_2(\mathcal{A}, \mathsf{Sim}_\mathcal{S}, x_\mathcal{R}, x_\mathcal{S})$$

where Expt_1 and Expt_2 are defined in Figure 3.

4 Non-Malleable Oblivious Transfer

4.1 Definition

We define *non-malleable* OT which considers a man-in-the-middle adversary that generates OT messages as a function of those generated by honest players. The

- The honest receiver $\mathcal{R}$ sends $x_{\mathcal{R}}$ and f to the ideal functionality.
- Initialize $\mathcal{A}$ with uniform random tape r.
- $\mathsf{Sim}^1_{\mathcal{R}}$ on input f, interacts with $\mathcal{A}$ and outputs $\pi_1, \pi_2, x_{\mathcal{S}}$ and sk.
- Send $x_{\mathcal{S}}$ to the ideal functionality.
- $\mathsf{Sim}^2_{\mathcal{R}}$ on input sk, interacts with $\mathcal{A}$ and outputs π_3 and π_4. $\mathsf{Sim}^2_{\mathcal{R}}$ may send an abort to the ideal functionality.
- Output $(r, \pi_1, \pi_2, \pi_3, \pi_4)$ and the output of the honest $\mathcal{R}$.

Fig. 2: Ideal Experiment in the Receiver Security Game

$\mathsf{Expt}_1(\mathcal{A}, \Pi, x_{\mathcal{S}}) = 1]$	$\mathsf{Expt}_2(\mathcal{A}, \mathsf{Sim}_{\mathcal{S}}, x_{\mathcal{S}})$
– Initialize $\mathcal{A}$ with a uniform random tape s.	– Initialize $\mathcal{A}$ with a uniform random tape s.
– $\pi_1 \leftarrow \mathcal{A}(1^\lambda; s)$.	– $\mathsf{Sim}^1_{\mathcal{S}}$ interacts with $\mathcal{A}$ and produces (π_1, sk).
– Choose $r \leftarrow \{0,1\}^\lambda$ uniformly at random and compute $\pi_2 \leftarrow \Pi_2(x_{\mathcal{S}}, \pi_1; r)$.	– $\mathsf{Sim}^2_{\mathcal{S}}$ on input sk interacts with $\mathcal{A}$ and produces a query $(x_{\mathcal{R}}, f_0, f_1)$ to be sent to the ideal functionality.
– $(f_0, \pi_3[0]), (f_1, \pi_3[1]) \leftarrow \mathcal{A}(\pi_2; s)$.	– On receiving $z_b = f_b(x_{\mathcal{R}}, x_{\mathcal{S}})$ from the ideal functionality, $\mathsf{Sim}^2_{\mathcal{S}}$ interacts with $\mathcal{A}$ and produces $(\pi_2, \{f_b, \pi_3[b], \pi_4[b]\}_{b\in\{0,1\}})$.
– $\pi_4[b] \leftarrow \Pi_4(x_{\mathcal{S}}, \pi_1, (f_b, \pi_3[b]); r)$ for $b \in \{0,1\}$.	
– Output $(s, \pi_1, \pi_2, \{f_b, \pi_3[b], \pi_4[b]\}_{b\in\{0,1\}})$.	– Output $(s, \pi_1, \pi_2, \{f_b, \pi_3[b], \pi_4[b]\}_{b\in\{0,1\}})$.

Fig. 3: Descriptions of Expt_1 and Expt_2.

non-malleability property ensures that no PPT adversarial sender can generate its OT inputs as a function of the (secret) inputs of honest senders.

Definition 4 (ℓ non-malleable $\binom{m}{k}$ Oblivious Transfer). *An ℓ non-malleable $\binom{m}{k}$ Oblivious Transfer is a protocol between a sender $\mathcal{S}$ with inputs $\{\mathsf{m}_i\}_{i\in[m]}$ and a receiver $\mathcal{R}$ with input $K \subset [m]$ where $|K| = k$, that satisfies the following:*

- **Correctness.** *For every $i \in [m], \mathsf{m}_i \in \{0,1\}^\lambda$ and $K \subset [m]$ such that $|K| = k$,*
$$\mathsf{Out}_{\mathcal{R}}\langle \mathcal{S}(\{\mathsf{m}_i\}_{i\in[m]}), \mathcal{R}(K)\rangle = \{m_i\}_{i\in K}$$
- **(Parallel Composable) Receiver Security.** *For every PPT sender $\mathcal{S}^*$ and every pair K, K' of k-sized subsets of $[m]$, we require*
$$\mathsf{View}_{\mathcal{S}^*}\langle \mathcal{S}^*, \mathcal{R}(K)\rangle \approx_c \mathsf{View}_{\mathcal{S}^*}\langle \mathcal{S}^*, \mathcal{R}(K')\rangle$$

Additionally, we require that there exists a PPT extractor $\mathsf{Sen.Ext}$ *that on input any transcript* τ *and with black-box access to any PPT sender* $\mathcal{S}^*$ *outputs* $\{(\mathsf{m}^*_{i,j})\}_{i\in[m],j\in[\ell]}$ *where* $\mathsf{m}^*_{i,j}$ *denotes the* i^{th} *implicit input used by* $\mathcal{S}^*$ *in the* j^{th} *session of* τ *(if any input is not well-defined, it outputs* $\perp$ *in its place)*[9].

– **Non-Malleability.** *Consider any PPT adversary (denoted by* MIM*) that interacts with upto* ℓ *senders* $\mathcal{S}_1,\ldots,\mathcal{S}_\ell$ *on the left, where for every* $j\in[\ell]$*,* $\mathcal{S}_j$ *has input* $\{\mathsf{m}_{i,j}\in\{0,1\}^n\}_{i\in[m]}$*, and upto* ℓ *receivers* $\mathcal{R}_1,\ldots,\mathcal{R}_\ell$ *on the right, where for every* $j\in[\ell]$*,* $\mathcal{R}_j$ *has input* K_j*.*
We denote by $\mathsf{View}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]},\{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle$ *the view of the* MIM *in this interaction, and denote the* i^{th} *implicit input used by the* MIM *in the* j^{th} *right session by* $\mathsf{m}'_{i,j}$ [10]*. We denote by* $\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]},\{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle$ *the joint distribution of* $\{(\mathsf{m}'_{i,j})\}_{i\in[m],j\in[\ell]}$ *and* $\mathsf{View}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]},\{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle$*. Then, we require that there exists a simulator-extractor pair,* $(\mathsf{Sim}_{\mathsf{OT}},\mathsf{Ext}_{\mathsf{OT}})$ *that outputs*

$$\mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]},\{K_j\}_{j\in[\ell]})=\mathsf{Sim}_{\mathsf{OT}}^{\mathsf{MIM},\{\mathsf{OT}(\{\mathsf{m}_{i,j}\}_{i\in[m]},\cdot)\}_{j\in[\ell]}}(\sigma,\{\widetilde{K}_j\}_{j\in[\ell]}),$$

for $(\sigma,\{\widetilde{K}_j\}_{j\in[\ell]})\leftarrow\mathsf{Ext}_{\mathsf{OT}}^{\mathsf{MIM}}(\{K_j\}_{j\in[\ell]})$*, s.t. for all honest inputs* $\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]}$*,* $\{K_j\}_{j\in[\ell]}$*, we have*

$$\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]},\{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle\approx_c\mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]},\{K_j\}_{j\in[\ell]}).$$

4.2 Construction

In this subsection, we construct ℓ non-malleable m-choose-k OT. Here, ℓ denotes the number of executions that an MIM adversary may participate in. Our construction is described in Figure 4, and makes use of the following:

– A 4 round two-party secure computation protocol Π with delayed-function selection and 1-rewinding sender security.
– An information-theoretic $m(\lambda)\cdot\ell(\lambda)$ non-malleable secret sharing scheme.
– A low-depth proof for P.
– An existentially unforgeable signature scheme with algorithms denoted by $\mathsf{Signature.Setup}$, $\mathsf{Signature.Sign}$ and $\mathsf{Signature.Verify}$.

We describe our protocol formally in Figure 4. The correctness of this protocol follows from correctness of the underlying oblivious transfer, non-malleable codes and signature scheme. In what follows, we prove security of this protocol.

Theorem 2. *Let* λ *denote the security parameter, and* $m=m(\lambda),k=k(\lambda),\ell=\ell(\lambda)$ *be arbitrary fixed polynomials. There exists a 4 round* ℓ *non-malleable* $\binom{m}{k}$ *oblivious transfer protocol satisfying Definition 4 that makes black-box use of any 4 round two-party secure computation protocol* Π *satisfying Definition 3, and any existentially unforgeable signature scheme.*

[9] This property guarantees parallel composability, and is satisfied by most natural rewinding-based protocols.
[10] If any of these is not well-defined, we denote it by $\perp$.

Inputs: Sender $\mathcal{S}$ has inputs $\{\mathsf{m}_j\}_{j\in m}$ and receiver $\mathcal{R}$ has input a set $K \subseteq [m]$ where $|K| = k$.

Protocol: $\mathcal{S}$ and $\mathcal{R}$ do the following.

1. $\mathcal{S}$ samples $(vk, sk) \leftarrow \mathsf{Signature.Setup}(1^\lambda)$, then does the following.
   - For each $i \in [\lambda], j \in [m]$, pick uniform randomness $r_{i,j}$ and compute $(\mathsf{L}_{i,j}, \mathsf{R}_{i,j}) = \mathsf{NM.Code}((vk|\mathsf{m}_j); r_{i,j})$.
   - Set $x = (vk, \{(\mathsf{L}_{i,j}, \mathsf{R}_{i,j}, \mathsf{m}_j)\}_{i\in[\lambda],j\in[m]})$ and $\mathcal{L} = \big\{(vk, \{(\mathsf{L}_{i,j}, \mathsf{R}_{i,j}, \mathsf{m}_j)\}_{i\in[\lambda],j\in[m]}) : \forall i \in [\lambda], j \in [m], \mathsf{NM.Decode}(\mathsf{L}_{i,j}, \mathsf{R}_{i,j}) = (vk|\mathsf{m}_j)\big\}$. Compute $\mathsf{ldp} = \mathsf{LDP.Prove}(x, \mathcal{L})$.
2. For each $i \in [\lambda]$, $\mathcal{R}$ picks $\mathsf{c}_i \leftarrow \{0, 1\}$.
3. Both parties engage in the protocol Π to compute functionality $\mathcal{F}$ where:
   - $\mathcal{R}$ plays the receiver with input K committed in round 1 and delayed function $(\mathsf{c}_1, \ldots, \mathsf{c}_\lambda)$ chosen in round 3.
   - $\mathcal{S}$ plays the sender with input (x, ldp), where x is parsed as $(vk, \{\mathsf{m}_j, (\mathsf{L}_{i,j}, \mathsf{R}_{i,j})\}_{i\in[\lambda],j\in[m]}$.
   - The functionality $\mathcal{F}$ on input $(vk, \{\mathsf{m}_j, \mathsf{L}_{i,j}, \mathsf{R}_{i,j}\}_{i\in[\lambda],j\in[m]}, K, \{\mathsf{c}_i\}_{i\in[\lambda]})$ generates an output as follows:
     - If $\mathsf{LDP.Verify}(x, \mathsf{ldp}) \neq 1$, output $\perp$.
     - Otherwise set $\mathsf{out} = vk, \{\mathsf{m}_j\}_{j\in K}$. Additionally, for every $i \in [\lambda]$, if $c_i = 0$, append $(\{\mathsf{L}_{i,j}\}_{j\in[m]})$ to out, else append $(\{\mathsf{R}_{i,j}\}_{j\in[m]})$ to out.
     - Output out.

     Additionally, $\mathcal{S}$ signs messages generated according to Π, denoted by (Π_2, Π_4). It sets $\sigma_2 = \mathsf{Signature.Sign}(\Pi_2, sk)$, $\sigma_4 = \mathsf{Signature.Sign}(\Pi_4, sk)$. It sends (σ_2, σ_4) to $\mathcal{R}$.
4. $\mathcal{R}$ obtains output out and parses $\mathsf{out} = (vk, \{\mathsf{m}_j\}_{j\in K}, \cdot)$. It outputs $\{\mathsf{m}_j\}_{j\in K}$ iff $\mathsf{Signature.Verify}(\sigma_2, \Pi_2, vk) \wedge \mathsf{Signature.Verify}(\sigma_4, \Pi_4, vk) = 1$, otherwise $\perp$.

Fig. 4: $\ell(\lambda)$ Non-Malleable $m(\lambda)$-choose-$k(\lambda)$ Oblivious Transfer

By relying on our 4 round two-party secure computation protocol satisfying Definition 3 based on black-box use of any public-key encryption with pseudo-random public keys, we obtain the following Corollary.

Corollary 1. *Let λ denote the security parameter, and $m = m(\lambda), k = k(\lambda), \ell = \ell(\lambda)$ be arbitrary polynomials. There exists a 4 round ℓ non-malleable $\binom{m}{k}$ OT protocol satisfying Definition 4 that makes black-box use of any public-key encryption with pseudo-random public keys.*

4.3 Security

We consider any man-in-the-middle adversary that participates as an OT receiver in upto $\ell(\lambda)$ executions of this protocol on the right, and participates as an OT sender in upto $\ell(\lambda)$ executions on the left.

We will prove that there exists a PPT algorithm $\mathsf{Sim\text{-}Ext}$, that with black-box access to the MIM, to ℓ copies of the ideal OT functionality $\mathbf{OT} = \{\mathsf{OT}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]}, \cdot)\}_{j\in[\ell]}$ and with input $\{K_j\}_{j\in[\ell]}$, simulates an execution of the protocol with the $\mathsf{MIM}$and extracts all the inputs $\{(\{\widetilde{\mathsf{m}}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]}$ used by the MIMin the executions where the MIM is sender. We will prove that $\mathsf{Sim\text{-}Ext}$ outputs $\mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]}, \{K_j\}_{j\in[\ell]})$ such that

$$\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]}, \{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle \approx_c \mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]}, \{K_j\}_{j\in[\ell]})$$

To prove indistinguishability, we define a sequence of hybrid experiments, where the first one outputs the distribution $\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]}, \{\mathcal{R}_j(K_j)\}_{j\in[\ell]}$ and the final one outputs the distribution $\mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_{i,j}\}_{i\in[m],j\in[\ell]}, \{K_j\}_{j\in[\ell]})$. Formally, these hybrids are defined as follows:

Hyb_0 : This corresponds to an execution of the $\mathsf{MIM}$with ℓ honest senders $\{\mathcal{S}_j\}_{j\in[\ell]}$ on the left, each using inputs $\{\mathsf{m}_{i,j}\}_{i\in[m]}$ respectively and ℓ honest receivers on the right with inputs $(\{K_j\}_{j\in[\ell]})$ respectively. The output of this hybrid is $\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]}, \{\mathcal{R}_j(K_j)\}_{j\in[\ell]}$.

Hyb_1 : This experiment modifies Hyb_1 by introducing an additional abort condition. Specifically, the experiment first executes the complete protocol corresponding to the real execution of the MIMexactly as in Hyb_0 to obtain the distribution $\mathsf{Real}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}_{i,j}\}_{i\in[m]})\}_{j\in[\ell]}, \{\mathcal{R}_j(K_j)\}_{j\in[\ell]}\rangle$.

Let $p(\lambda)$ denote the probability that the MIMcompletes this execution without aborting. Set $\gamma(\lambda) = \max\big(\lambda, p^{-2}(\lambda)\big)$. With the first two rounds of the transcript fixed, the rewind the right execution up to $\gamma(\lambda)$ times, picking inputs $(\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\lambda)$ for each of the ℓ receivers $\{\mathcal{R}_j\}_{j\in[\ell]}$ independently and uniformly at random in every run. If there exists a rewinding thread where the MIM completes the protocol execution, denote the inputs chosen by the challenger on behalf of the honest receiver in this rewinding thread by $(\mathsf{c'}^j_1, \ldots, \mathsf{c'}^j_\lambda)$. For every $j \in [\ell]$, let index $\alpha_j \in [\lambda]$ be such that $\mathsf{c}^j_{\alpha_j} = 0$ and $\mathsf{c'}^j_{\alpha_j} = 1$. Additionally for every $j \in [\ell], i \in [m]$, use $(\widetilde{\mathsf{L}}^j_{\alpha_j,i}, \widetilde{\mathsf{R}}^j_{\alpha_j,i})$ obtained as output from the main and rewinding executions respectively to compute $\widetilde{\mathsf{m}}^j_i = \mathsf{NM.Decode}(\widetilde{\mathsf{L}}^j_{\alpha_j,i}, \widetilde{\mathsf{R}}^j_{\alpha_j,i})$.

If no such rewinding thread exists, or if there exists $j \in [\ell]$ for which there does not exist $\alpha \in [\lambda]$ such that $\mathsf{c}^j_\alpha = 0$ and $\mathsf{c'}^j_\alpha = 1$, then set $\widetilde{\mathsf{m}}^j_i = \perp$ for all $i \in [m]$. Now, the output of this hybrid is the joint distribution

$$\mathsf{View}_{\mathsf{MIM}}\langle\{\mathcal{S}_j(\{\mathsf{m}^j_i\}_{i\in[m]})\}_{j\in[\ell]}, \{\mathcal{R}^j(K^j)\}_{j\in[\ell]}\rangle, \{\widetilde{\mathsf{m}}^j_i\}_{j\in[\ell],i\in[m]}.$$

Lemma 1. *For every unbounded distinguisher $\mathcal{D}$ we have*

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

Proof. Since the MIM's inputs $\{(\widetilde{\mathsf{m}}^j_i\}_{j\in[\ell]}$ are committed in round 2 of the protocol, then conditioned on the adversary providing a non-aborting transcript in

a rewinding execution in Hyb_1, by simulation security of the $\mathsf{2pc}$, $\{(\widetilde{\mathsf{m}}_i^j\}_{j\in[\ell]}$ are correctly extracted.

Therefore, to prove this lemma it suffices to show that such a rewinding execution (with a non-aborting transcript) can be found within $\gamma(\lambda)$ attempts, except with probability $\mathsf{negl}(\lambda)$. To see this, we observe that the probability of a non-aborting transcript is $p(\lambda)$, and therefore, the probability that all $\gamma(\lambda)$ trials abort is $(1-p(\lambda))^{\ell(\lambda)} \leq \exp^{p^{-1}(\lambda)} = \mathsf{negl}(\lambda)$.

Hyb_2: This experiment modifies Hyb_2 to execute the simulator of Π in all sessions where the MIM is a receiver. Specifically, in these executions, instead of the honest sender strategy with input $\{\mathsf{m}_i^j\}_{i\in[m],j\in[\ell]}$, we execute the simulator $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}^{\mathsf{MIM},\mathcal{F}(\mathsf{inp}_{\mathcal{S}^j},\cdot)}$ where

$$\mathsf{inp}_{\mathcal{S}^j} = (\{\mathsf{m}_i^j, \mathsf{L}_{1,i}^j, \ldots, \mathsf{L}_{\lambda,i}^j, \mathsf{R}_{1,i}^j, \ldots, \mathsf{R}_{\lambda,i}^j\}_{i\in[m]}).$$

$\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ expects round 1 and round 3 messages from the MIM, and the MIM in turn expects corresponding messages from the receiver in the right execution. Receiver messages for the right execution are generated using honest receiver strategy with inputs K^j fixed, and inputs $c_1^j, \ldots, c_\lambda^j$ chosen uniformly at random, exactly as in Hyb_1. Denote the view of the $\mathsf{MIM}$by

$$\mathsf{View}_{\mathsf{Sim}^{\{\mathcal{F}(\mathsf{inp}_{\mathcal{S}^j},\cdot)\}_{j\in[\ell]}}}\langle\{\mathcal{R}^j(K^j)\}_{j\in[\ell]}\rangle,$$

where for every $j \in [\ell]$, $\mathsf{inp}_{\mathcal{S}^j}$ is as defined above.

Next, with the first two rounds of the transcript fixed, the challenger rewinds the right execution up to $\ell(\lambda)$ times, picking inputs $(\mathsf{c}_1^j, \ldots, \mathsf{c}_\lambda^j)$ for $\mathcal{R}^j$ independently and uniformly at random in every run, and generating messages in the left execution by running the simulator $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ each time.

If there exists a rewinding execution where the MIMcompletes the protocol, denote the inputs chosen by the challenger on behalf of the honest receiver in this rewinding thread by $(\mathsf{c'}_1^j, \ldots, \mathsf{c'}_\lambda^j)$. For every $j \in [\ell]$, let index $\alpha_j \in [\lambda]$ be such that $\mathsf{c}_{\alpha_j}^j = 0$ and $\mathsf{c'}_{\alpha_j}^j = 1$. Additionally for every $j \in [\ell], i \in [m]$, use $(\widetilde{\mathsf{L}}_{\alpha_j,i}^j, \widetilde{\mathsf{R}}_{\alpha_j,i}^j)$ obtained as output from the main and rewinding executions respectively to compute $\widetilde{\mathsf{m}}_i^j = \mathsf{NM.Decode}(\widetilde{\mathsf{L}}_{\alpha_j,i}^j, \widetilde{\mathsf{R}}_{\alpha_j,i}^j)$. If no such rewinding thread exists, or if there exists $j \in [\ell]$ for which there does not exist $\alpha \in [\lambda]$ such that $\mathsf{c}_\alpha^j = 0$ and $\mathsf{c'}_\alpha^j = 1$, then set $\widetilde{\mathsf{m}}_i^j = \perp$ for all $i \in [m]$. The output of this hybrid is the joint distribution:

$$\mathsf{View}_{\mathsf{Sim}^{\{\mathcal{F}(\mathsf{inp}_{\mathcal{S}^j},\cdot)\}_{j\in[\ell]}}}\langle\{\mathcal{R}^j(K^j)\}_{j\in[\ell]}\rangle, \{\widetilde{\mathsf{m}}_i^j\}_{j\in[\ell],i\in[m]},$$

where for every $j \in [\ell]$, $\mathsf{inp}_{\mathcal{S}^j}$ is as defined above.

Lemma 2. *Assuming* 1*-rewinding secure two party computation, for every efficient distinguisher $\mathcal{D}$ we have*

$$\Big|\Pr[\mathcal{D}(\mathsf{Hyb}_1) = 1] - \Pr[\mathcal{D}(\mathsf{Hyb}_2) = 1]\Big| = \mathsf{negl}(\lambda)$$

Proof. We consider a sequence of sub-hybrids $\mathsf{Hyb}_{1,0}, \mathsf{Hyb}_{1,1}, \ldots \mathsf{Hyb}_{1,\ell}$ where for every $j \in [\ell]$, $\mathsf{Hyb}_{1,j}$ is identical to $\mathsf{Hyb}_{1,j-1}$, except that instead of executing the honest sender strategy using honest sender inputs $\{m_i^j\}_{i\in[m]}$, we execute the simulator in the j^{th} left execution, where $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}^{\mathsf{MIM},\mathcal{F}(\mathsf{inp}_{\mathcal{S}^j},\cdot)}$ where

$$\mathsf{inp}_{\mathcal{S}^j} = (\{\mathsf{m}_i^j, \mathsf{L}_{1,i}^j, \ldots, \mathsf{L}_{\lambda,i}^j, \mathsf{R}_{1,i}^j, \ldots, \mathsf{R}_{\lambda,i}^j\}_{i\in[m]})$$

Suppose the lemma is not true. Then for every large enough $\lambda \in \mathbb{N}$ there exists $j^*(\lambda) \in [\ell(\lambda)]$, a polynomial $p(\cdot)$ and a distinguisher $\mathcal{D}$ such that for infinitely many $\lambda \in \mathbb{N}$,

$$\left|\Pr[\mathcal{D}(\mathsf{Hyb}_{1,j^*-1}) = 1] - \Pr[\mathcal{D}(\mathsf{Hyb}_{1,j^*}) = 1]\right| = \frac{1}{q(\lambda)}$$

We derive a contradiction by building a reduction $\mathcal{A}$ that on input λ, obtains $j^*(\lambda)$ as advice and with black-box access to the $\mathsf{MIM}$and to $\mathcal{D}$ contradicts 1-rewinding security of the two party computation protocol. $\mathcal{A}$ proceeds as follows:

- $\mathcal{A}$ first creates receiver $\mathcal{R}'$ that interacts with the external challenger as follows.
  - Generate the first round messages according to receiver strategy with inputs $\{K^j\}_{j\in[\ell]}$ for the right execution. Obtain first round messages from the MIM, and output the MIM's message in the j^{*th} left execution to the challenger of the $\mathsf{2pc}$.
  - Obtain the second round message for the left execution externally from the $\mathsf{2pc}$ challenger, and forward this to the MIMas $\mathcal{S}^{j^*}$'s message in the j^{*th} left execution. Obtain the second round message from the MIM for the right execution.
  - Generate the third round message for the right execution according to honest receiver strategy, and obtain the third round message from the MIM. Output the MIM's message in left session j^* to the challenger.
  - Obtain the fourth round message for the left execution externally from the challenger, and forward this to the MIMas $\mathcal{S}$'s message in the j^{*th} left execution. Obtain the fourth round message from the $\mathsf{MIM}$for the right execution.
- Next, $\mathcal{A}$ rewinds $\mathcal{R}'$ once, as follows.
  - Generate the third round message according to honest receiver strategy, and obtain the third round message from the MIM. Output the MIM's message in session j^* to the challenger.
  - Obtain the fourth round message for the left execution externally from the challenger, and forward this to the MIMas $\mathcal{S}$'s message in the j^{*th} left execution. Obtain the fourth round message from the $\mathsf{MIM}$from the left execution.
  - If none of the executions abort, for every $j \in [\ell]$, find $\alpha_j \in [\lambda]$ such that $\mathsf{c}_{\alpha_j}^j = 0$ and $\mathsf{c}'^j_{\alpha_j} = 1$. and use it to compute $\widetilde{\mathsf{m}}_i^j = \mathsf{NM.Decode}(\widetilde{\mathsf{L}}_{\alpha_j,i}^j, \widetilde{\mathsf{R}}_{\alpha_j,i}^j)$ for $i \in [m], j \in [\ell]$. Else, set $\widetilde{\mathsf{m}}_i^j = \perp$ for $i \in [m], j \in [\ell]$

– $\mathcal{A}$ outputs the entire view of $\mathcal{R}'$ together with $\{\widetilde{\mathsf{m}}_i^j\}_{i\in[m],j\in[\ell]}$. If the challenger used honest sender messages, we denote the distribution output by $\mathcal{A}$ in this experiment by Dist_1 and if the challenger used simulated messages, we denote the distribution output by $\mathcal{A}$ in this experiment by Dist_2.

If the challenger's messages correspond to the real sender $\mathcal{S}$, then the distribution output by $\mathcal{A}$ conditioned on not aborting corresponds to Hyb_1, and if the challenger's messages correspond to $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$, then the distribution output by $\mathcal{A}$ conditioned on not aborting corresponds to Hyb_2.

By assumption, for infinitely many $\lambda \in \mathbb{N}$,

$$\left|\Pr[\mathcal{D}(\mathsf{Hyb}_1) = 1] - \Pr[\mathcal{D}(\mathsf{Hyb}_2) = 1]\right| = \frac{1}{q(\lambda)}$$

Since the MIM completes any run of the protocol without aborting with probability at least $p(\lambda)$, and because aborts are independent of the distinguishing advantage, for infinitely many $\lambda \in \mathbb{N}$:

$$\left|\Pr[\mathcal{D} = 1 \ \wedge \ \neg\mathsf{abort}|\mathsf{Hyb}_1] - \Pr[\mathcal{D} = 1 \ \wedge \ \neg\mathsf{abort}|\mathsf{Hyb}_2]\right| \geq \frac{1}{p(\lambda)\cdot q(\lambda)}$$

where $\neg\mathsf{abort}$ denotes the event that an execution that is completed in the main thread, is also completed without aborting in one rewinding execution.

This implies that for infinitely many $\lambda \in \mathbb{N}$:

$$\left|\Pr[\mathcal{D}(\mathsf{Dist}_1) = 1] - \Pr[\mathcal{D}(\mathsf{Dist}_2) = 1]\right| \geq \frac{1}{p(\lambda)\cdot q(\lambda)},$$

and thus $\mathcal{D}$ contradicts 1-rewinding security of the two party computation protocol.

Hyb_3: This hybrid is the same as Hyb_2 except whenever the challenger obtains as output a verification key in one of the right sessions that is identical to a verification key used in one of the left sessions, the hybrid outputs $\perp$. By existential unforgeability of the signature scheme, given any PPT adversary MIM, Hyb_2 and Hyb_3 are statistically indistinguishable.

Hyb_4: This hybrid is the same as Hyb_3 except that $\mathsf{inp}_{\mathcal{S}^j}$ is set differently. Specifically, for every $j \in [\ell], i \in [m]$ and $\alpha \in [\lambda]$, we set $(\mathsf{L}^j_{\alpha,i}, \mathsf{R}^j_{\alpha,i}) \leftarrow \mathsf{NM.Sim}(1^{p(\lambda)})$, and set

$$\mathsf{inp}_{\mathcal{S}^j} = (\{\mathsf{m}_i^j, \mathsf{L}^j_{1,i}, \ldots, \mathsf{L}^j_{\lambda,i}, \mathsf{R}^j_{1,i}, \ldots, \mathsf{R}^j_{\lambda,i}\}_{i\in[m]}).$$

We note that at this point, the functionality $\{\mathcal{F}(\mathsf{inp}_{\mathcal{S}^j}, \cdot)\}_{j\in[\ell]}$ can be perfectly simulated with access to the ideal functionality $\{\mathsf{OT}^j(\mathsf{m}_i^j, \mathsf{m}_i^j, \cdot)\}_{j\in[\ell]}$. Therefore, the output of this hybrid is identical to the ideal view $\mathsf{Ideal}_{\mathsf{MIM}}(\{\mathsf{m}_i^j\}_{i\in[m],j\in[\ell]}, \{K^j\}_{j\in[\ell]})$.

Lemma 3. *Assuming $m(\lambda)\cdot\ell(\lambda)$ symmetric non-malleable codes, for every unbounded distinguisher $\mathcal{D}$ we have:*

$$\left|\Pr[\mathcal{D}(\mathsf{Hyb}_4) = 1] - \Pr[\mathcal{D}(\mathsf{Hyb}_3) = 1]\right| = \mathsf{negl}(\lambda)$$

Proof. We prove indistinguishability between Hyb_3 and Hyb_4 by considering a sequence of sub-hybrids, $\{\mathsf{Hyb}_{3,i,j,k}\}_{i\in[1,m],j\in[1,\ell],k\in[0,\lambda]}$ where:

- $\mathsf{Hyb}_3 = \mathsf{Hyb}_{3,0,\ell,\lambda}$, $\mathsf{Hyb}_4 = \mathsf{Hyb}_{3,m,\ell,\lambda}$,
- for $i \in [m]$, $\mathsf{Hyb}_{3,i-1,\ell,\lambda} = \mathsf{Hyb}_{3,i,1,0}$
- for $j \in [\ell]$, $\mathsf{Hyb}_{3,i,j-1,\lambda} = \mathsf{Hyb}_{3,i,j,0}$,
- for every $i \in [m], j \in [\ell], k \in [\lambda]$, $\mathsf{Hyb}_{3,i,j,k}$ is identical to $\mathsf{Hyb}_{3,i,j,k-1}$ except that $\mathsf{Hyb}_{3,i,j,k}$ samples $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i}) \leftarrow \mathsf{NM.Code}(0)$.

Suppose the lemma is not true. Then there exists $i^* \in [m], j^* \in [\ell], k^* \in [\lambda]$, an unbounded distinguisher $\mathcal{D}$ and a polynomial $p(\cdot)$ such that for large enough $\lambda \in \mathbb{N}$,

$$\left|\Pr[\mathcal{D}(\mathsf{Hyb}_{3,i^*,j^*,k^*}) = 1] - \Pr[\mathcal{D}(\mathsf{Hyb}_{3,i^*,j^*,k^*-1}) = 1]\right| = \frac{1}{p(\lambda)} \tag{1}$$

We now define a pair of tampering functions $(f_{\mathsf{MIM}}, g_{\mathsf{MIM}})$, and additional function h_{MIM} as follows:

- $f_{\mathsf{MIM}}, g_{\mathsf{MIM}}$ and h_{MIM} share common state that is generated as follows:
  - Execute $\mathsf{Sim\text{-}2PC}^{\mathsf{MIM}}_{\mathsf{Sen}}$, using honest $\mathcal{R}$ strategy in the right executions with input $\{K^j\}_{j\in[\ell]}$ and uniformly chosen $\{c^j_1, \ldots c^j_\lambda\}_{j\in[\ell]}$, until $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ generates a query to the ideal functionality $\mathcal{F}$ at the end of round 3.
  - At this point, $\mathsf{Sim\text{-}2PC}^{\mathsf{MIM}}_{\mathsf{Sen}}$ outputs a view and transcript of the MIMuntil the third round, as well as $\{\widetilde{K}^j, \widetilde{\mathsf{c}}^j_1, \ldots, \widetilde{\mathsf{c}}^j_\lambda\}_{j\in[\ell]}$ that correspond to the receiver's inputs in the left execution.
  - Rewind the third round once with uniformly and independently chosen $\{\mathsf{c'}^j_1, \ldots, \mathsf{c'}^j_\lambda\}_{j\in[\ell]}$. If for every $j \in [\ell(\lambda)]$, there exists $\alpha_j \in [\lambda]$ such that $\mathsf{c}^j_{\alpha_j} = 0$ and $\mathsf{c'}^j_{\alpha_j} = 1$, continue, otherwise abort.
  - Obtain the rewinding view (with the same prefix of the first two rounds), as well as $(\overline{\mathsf{c}}_1, \ldots, \overline{\mathsf{c}}_n)$ that correspond to the receiver's input in the left session in this rewinding execution. If $\widetilde{\mathsf{c}}^j_k \neq \overline{\mathsf{c}}^j_k$, continue. Otherwise, abort.
  - Generate $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i})$ for every $(i,j,k) \in [m] \times [\ell] \times [\lambda] \setminus \{i^*, j^*, k^*\}$ according to $\mathsf{Hyb}_{3,i^*,j^*,k^*-1}$ (this is identical to setting them according to $\mathsf{Hyb}_{3,i^*,j^*,k^*}$).
  - Output the view of the MIMuntil round 3 in the main the rewinding threads, including (i^*, j^*, k^*), the values $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i})_{(i,j,k)\in[m]\times[\ell]\times[\lambda]\setminus\{i^*,j^*,k^*\}}$.
  - Additionally, output the receiver's inputs $\{\widetilde{K}^j, \widetilde{\mathsf{c}}^j_1, \ldots, \widetilde{\mathsf{c}}^j_\lambda\}_{j\in[\ell]}$ and the sender's inputs $\{sk^j, vk^j, \{\mathsf{m}^j_i\}_{i\in[m]}\}_{j\in[\ell]}$.
- Next, the function h_{MIM} on input L, sets $\mathsf{L}^{j^*}_{k^*,i^*} = \mathsf{L}, \mathsf{R}^{j^*}_{k^*,i^*} = 0$.

  Now, using hardwired values $\{vk^j, \{\mathsf{m}^j_i\}_{i\in[m]}\}_{j\in[\ell]}$, $\{\widetilde{K}^j, \widetilde{\mathsf{c}}^j_1, \ldots, \widetilde{\mathsf{c}}^j_\lambda\}_{j\in[\ell]}$ as well as the values $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i})_{(i,j,k)\in[m]\times[\ell]\times[\lambda]\setminus\{i^*,j^*,k^*\}}$, it computes

$$\mathsf{out} = \{\mathcal{F}^j(vk^j, \{\mathsf{m}_i, \mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i}\}_{i\in[m],k\in[\lambda]}, \widetilde{K}^j, \{\widetilde{\mathsf{c}}^j_k\}_{k\in[\lambda]})\}_{j\in[\ell]}.$$

It then invokes $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ on out to generate the fourth round message of the protocol transcript in the main thread if $\widetilde{\mathsf{c}}^{j^*}_{k^*} = 0$, and generates the fourth round message of the protocol transcript in the rewinding thread if $\bar{\mathsf{c}}^{j^*}_{k^*} = 0$. It outputs the resulting transcript as the view of the MIM.

– The function f_{MIM} on input L, sets $\mathsf{L}^{j^*}_{k^*,i^*} = \mathsf{L}, \mathsf{R}^{j^*}_{k^*,i^*} = 0$.
Now, using hardwired values $\{vk^j, \{\mathsf{m}^j_i\}_{i\in[m]}\}_{j\in[\ell]}$, $\{\widetilde{K}^j, \widetilde{\mathsf{c}}^j_1, \ldots, \widetilde{\mathsf{c}}^j_\lambda\}_{j\in[\ell]}$ as well as the values $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i})_{(i,j,k)\in[m]\times[\ell]\times[\lambda]\setminus\{i^*,j^*,k^*\}}$, it computes

$$\mathsf{out} = \{\mathcal{F}^j vk^j, \{\mathsf{m}_i, \mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i}\}_{i\in[m],k\in[\lambda]}, \widetilde{K}^j, \{\widetilde{\mathsf{c}}^j_k\}_{k\in[\lambda]})\}_{j\in[\ell]}.$$

It then invokes $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ on out to generate the fourth round message of the protocol transcript in the main thread if $\widetilde{\mathsf{c}}^{j^*}_{k^*} = 0$, and generates the fourth round message of the protocol transcript in the rewinding thread if $\bar{\mathsf{c}}^{j^*}_{k^*} = 0$. It outputs the values $\{\mathsf{L}^j_{\alpha_j,i}\}_{i\in[m],j\in[\ell]}$ or $\{\mathsf{R}^j_{\alpha_j,i}\}_{i\in[m],j\in[\ell]}$ obtained from the MIM.

– The function g_{MIM} on input R, sets $\mathsf{R}^{j^*}_{k^*,i^*} = \mathsf{R}, \mathsf{L}^{j^*}_{k^*,i^*} = 0$.
Now, using hardwired values $\{vk^j, \{\mathsf{m}^j_i\}_{i\in[m]}\}_{j\in[\ell]}$, $\{\widetilde{K}^j, \widetilde{\mathsf{c}}^j_1, \ldots, \widetilde{\mathsf{c}}^j_\lambda\}_{j\in[\ell]}$ as well as the values $(\mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i})_{(i,j,k)\in[m]\times[\ell]\times[\lambda]\setminus\{i^*,j^*,k^*\}}$, it computes

$$\mathsf{out} = \{\mathcal{F}^j vk^j, \{\mathsf{m}_i, \mathsf{L}^j_{k,i}, \mathsf{R}^j_{k,i}\}_{i\in[m],k\in[\lambda]}, \widetilde{K}^j, \{\widetilde{\mathsf{c}}^j_k\}_{k\in[\lambda]})\}_{j\in[\ell]}.$$

It then invokes $\mathsf{Sim\text{-}2PC}_{\mathsf{Sen}}$ on out to generate the fourth round message of the protocol transcript in the main thread if $\widetilde{\mathsf{c}}^{j^*}_{k^*} = 1$, and generates the fourth round message of the protocol transcript in the rewinding thread if $\bar{\mathsf{c}}^{j^*}_{k^*} = 1$. It outputs the values $\{\mathsf{L}^j_{\alpha_j,i}\}_{i\in[m],j\in[\ell]}$ or $\{\mathsf{R}^j_{\alpha_j,i}\}_{i\in[m],j\in[\ell]}$ obtained from the MIM.

By security augmented non-malleable codes,

$$\Big(\mathsf{L}, \mathsf{NM.Decode}\big(f_{\mathsf{MIM}}(\mathsf{L}), g_{\mathsf{MIM}}(\mathsf{R})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(\mathsf{m}^{j^*}_{i^*}))\Big) \approx_\epsilon$$
$$\Big(\mathsf{L}, \mathsf{NM.Decode}\big(f_{\mathsf{MIM}}(\mathsf{L}), g_{\mathsf{MIM}}(\mathsf{R})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(0))\Big) \text{ and}$$
$$\Big(\mathsf{L}, \mathsf{NM.Decode}\big(g_{\mathsf{MIM}}(\mathsf{R}), f_{\mathsf{MIM}}(\mathsf{L})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(\mathsf{m}^{j^*}_{i^*}))\Big) \approx_\epsilon$$
$$\Big(\mathsf{L}, \mathsf{NM.Decode}\big(g_{\mathsf{MIM}}(\mathsf{R}), f_{\mathsf{MIM}}(\mathsf{L})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(0))\Big)$$

By the data processing inequality, this implies that for every function $h(\cdot)$,

$$\Big(h(\mathsf{L}), \mathsf{NM.Decode}\big(f_{\mathsf{MIM}}(\mathsf{L}), g_{\mathsf{MIM}}(\mathsf{R})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(\mathsf{m}^{j^*}_{i^*}))\Big) \approx_\epsilon$$
$$\Big(h(\mathsf{L}), \mathsf{NM.Decode}\big(f_{\mathsf{MIM}}(\mathsf{L}), g_{\mathsf{MIM}}(\mathsf{R})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(0))\Big) \text{ and}$$
$$\Big(h(\mathsf{L}), \mathsf{NM.Decode}\big(g_{\mathsf{MIM}}(\mathsf{R}), f_{\mathsf{MIM}}(\mathsf{L})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(\mathsf{m}^{j^*}_{i^*}))\Big) \approx_\epsilon$$
$$\Big(h(\mathsf{L}), \mathsf{NM.Decode}\big(g_{\mathsf{MIM}}(\mathsf{R}), f_{\mathsf{MIM}}(\mathsf{L})\big)\Big|(\mathsf{L}, \mathsf{R} \leftarrow \mathsf{NM.Code}(0))\Big)$$

Setting $h = h_{\mathsf{MIM}}$, for f_{MIM} and g_{MIM} defined above, these distributions correspond exactly to the outputs of $\mathsf{Hyb}_{3,i^*,j^*,k^*-1}$ and $\mathsf{Hyb}_{3,i^*,j^*,k^*}$ respectively, whenever $\widetilde{c}_{k^*}^{j^*} \neq \overline{c}_{k^*}^{j^*}$. Whenever $\widetilde{c}_{k^*}^{j^*} = \overline{c}_{k^*}^{j^*}$, the distributions $\mathsf{Hyb}_{3,i^*,j^*,k^*-1}$ and $\mathsf{Hyb}_{3,i^*,j^*,k^*}$ are statistically indistinguishable because they jointly only depend on one of the shares, L or R. Since $\epsilon(\lambda) = \mathsf{negl}(\lambda)$, this contradicts Equation (1), completing our proof.

5 Summary of Results

In this section, we provide a theorem statement capturing two of our main results. See full version for the proof.

Theorem 3. *Let f be an arbitrary multiparty functionality.*

- *In the watchlist correlations model, assuming black-box access to a pseudorandom generator, there exists a two-round protocol that computes f against static, malicious adversaries satisfying security with selective abort. For f in $\mathsf{NC1}$, a similar protocol exists unconditionally.*
- *Further assuming black-box access to a public-key encryption with pseudorandom public keys and a two-round oblivious transfer with semi-malicious security, there exists a protocol that securely computes f in five rounds in the plain model against static, malicious corruptions of all-but-one players satisfying security with selective abort.*

The communication and computation costs of both the protocols are $\mathsf{poly}(\lambda, n, |f|)$, where $|f|$ denotes the size of the circuit computing f, and where communication is over a broadcast channel.

Acknowledgements. Y. Ishai was supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, BSF grant 2018393, and ISF grant 2774/20. D. Khurana was supported from a DARPA SIEVE award. A. Sahai was supported in part from a DARPA SIEVE award, NTT Research, NSF Frontier Award 1413955, BSF grant2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through Award HR00112020024. Work done in part when A. Srinivasan was at UC Berkeley and supported in part by AFOSR Award FA9550-19-1-0200, AFOSR YIP Award, NSF CNS Award 1936826, DARPA/ARL SAFEWARE Award W911NF15C0210, a Hellman Award and research grants by the Sloan Foundation, Okawa Foundation, Visa Inc., and Center for Long-Term Cybersecurity (CLTC, UC Berkeley). The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies.

References

1. Afshar, A., Mohassel, P., Pinkas, B., Riva, B.: Non-interactive secure computation based on cut-and-choose. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 387–404. Springer, Heidelberg (May 2014)
2. Aggarwal, D., Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Optimal computational split-state non-malleable codes. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 393–417. Springer, Heidelberg (Jan 2016)
3. Aggarwal, D., Damgård, I., Nielsen, J.B., Obremski, M., Purwanto, E., Ribeiro, J., Simkin, M.: Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 510–539. Springer, Heidelberg (Aug 2019)
4. Aggarwal, D., Dziembowski, S., Kazana, T., Obremski, M.: Leakage-resilient non-malleable codes. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 398–426. Springer, Heidelberg (Mar 2015)
5. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 2087–2104. ACM Press (Oct / Nov 2017)
6. Ananth, P., Choudhuri, A.R., Jain, A.: A new approach to round-optimal secure multiparty computation. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 468–499. Springer, Heidelberg (Aug 2017)
7. Applebaum, B., Brakerski, Z., Garg, S., Ishai, Y., Srinivasan, A.: Separating two-round secure computation from oblivious transfer. In: Vidick, T. (ed.) 11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA. LIPIcs, vol. 151, pp. 71:1–71:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
8. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (Apr 2012)
9. Badrinarayanan, S., Goyal, V., Jain, A., Kalai, Y.T., Khurana, D., Sahai, A.: Promise zero knowledge and its applications to round optimal MPC. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 459–487. Springer, Heidelberg (Aug 2018)
10. Badrinarayanan, S., Goyal, V., Jain, A., Khurana, D., Sahai, A.: Round optimal concurrent MPC via strong simulation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 743–775. Springer, Heidelberg (Nov 2017)
11. Beaver, D.: Correlated pseudorandomness and the complexity of private computations. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 479–488. ACM (1996), https://doi.org/10.1145/237814.237996
12. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: STOC. pp. 503–513 (1990)
13. Benhamouda, F., Lin, H.: k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 500–532. Springer, Heidelberg (Apr / May 2018)

14. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: Silent OT extension and more. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 489–518. Springer, Heidelberg (Aug 2019)
15. Brakerski, Z., Halevi, S., Polychroniadou, A.: Four round secure computation without setup. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 645–677. Springer, Heidelberg (Nov 2017)
16. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Proceedings of the 2017 acm sigsac conference on computer and communications security. pp. 1825–1842 (2017)
17. Chase, M., Dodis, Y., Ishai, Y., Kraschewski, D., Liu, T., Ostrovsky, R., Vaikuntanathan, V.: Reusable non-interactive secure computation. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 462–488. Springer, Heidelberg (Aug 2019)
18. Chattopadhyay, E., Goyal, V., Li, X.: Non-malleable extractors and codes, with their many tampered extensions. In: Wichs, D., Mansour, Y. (eds.) 48th ACM STOC. pp. 285–298. ACM Press (Jun 2016)
19. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Round optimal secure multiparty computation from minimal assumptions. In: TCC 2020, Part II. pp. 291–319 (2020)
20. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 270–299. Springer, Heidelberg (Aug 2016)
21. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 127–157. Springer, Heidelberg (Aug 2017)
22. Damgård, I., Ishai, Y.: Constant-round multiparty computation using a black-box pseudorandom generator. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 378–394. Springer, Heidelberg (Aug 2005)
23. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. J. ACM 65(4), 20:1–20:32 (2018), https://doi.org/10.1145/3178432
24. Friolo, D., Masny, D., Venturi, D.: A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. Lecture Notes in Computer Science, vol. 11891, pp. 111–130. Springer (2019), https://doi.org/10.1007/978-3-030-36030-6_5
25. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS. pp. 40–49. IEEE Computer Society Press (Oct 2013)
26. Garg, S., Ishai, Y., Srinivasan, A.: Two-round MPC: information-theoretic and black-box. In: Beimel, A., Dziembowski, S. (eds.) Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11239, pp. 123–151. Springer (2018), https://doi.org/10.1007/978-3-030-03807-6_5
27. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: EUROCRYPT. pp. 448–476 (2016)
28. Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 468–499. Springer, Heidelberg (Apr / May 2018)

29. Giacomelli, I., Madsen, J., Orlandi, C.: Zkboo: Faster zero-knowledge for boolean circuits. In: 25th {usenix} security symposium ({usenix} security 16). pp. 1069–1083 (2016)
30. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. 25(1), 169–192 (1996)
31. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987)
32. Goyal, V.: Constant round non-malleable protocols using one way functions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 695–704. ACM Press (Jun 2011)
33. Goyal, V., Jain, A., Khurana, D.: Witness signatures and non-malleable multi-prover zero-knowledge proofs. IACR Cryptology ePrint Archive 2015, 1095 (2015), http://eprint.iacr.org/2015/1095
34. Goyal, V., Kumar, A., Park, S., Richelson, S., Srinivasan, A.: Non-malleable commitments from non-malleable extractors. Manuscript, accessed via personal communication (2018)
35. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: 53rd FOCS. pp. 51–60. IEEE Computer Society Press (Oct 2012)
36. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: STOC. pp. 1128–1141 (2016)
37. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: 55th FOCS. pp. 41–50. IEEE Computer Society Press (Oct 2014)
38. Goyal, V., Srinivasan, A., Zhu, C.: Multi-source non-malleable extractors and applications. Cryptology ePrint Archive, Report 2020/157 (2020), https://eprint.iacr.org/2020/157
39. Haitner, I., Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions of protocols for secure computation. SIAM J. Comput. 40(2), 225–266 (2011), https://doi.org/10.1137/100790537
40. Halevi, S., Hazay, C., Polychroniadou, A., Venkitasubramaniam, M.: Round-optimal secure multi-party computation. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 488–520. Springer, Heidelberg (Aug 2018)
41. Hazay, C., Ishai, Y., Marcedone, A., Venkitasubramaniam, M.: Leviosa: Lightweight secure arithmetic computation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) CCS 2019. pp. 327–344. ACM (2019), https://doi.org/10.1145/3319535.3354258
42. Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Actively secure garbled circuits with constant communication overhead in the plain model. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part II. LNCS, vol. 10678, pp. 3–39. Springer, Heidelberg (Nov 2017)
43. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 8–26. Springer, Heidelberg (Aug 1990)
44. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (Aug 2003)
45. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 406–425. Springer, Heidelberg (May 2011)

46. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) 39th ACM STOC. pp. 21–30. ACM Press (Jun 2007)
47. Ishai, Y., Kushilevitz, E., Paskin, A.: Secure multiparty computation with minimal interaction. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 577–594. Springer, Heidelberg (Aug 2010)
48. Ishai, Y., Kushilevitz, E., Prabhakaran, M., Sahai, A., Yu, C.: Secure protocol transformations. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. Lecture Notes in Computer Science, vol. 9815, pp. 430–458. Springer (2016), https://doi.org/10.1007/978-3-662-53008-5_15
49. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (Aug 2008)
50. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: Reingold, O. (ed.) Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5444, pp. 294–314. Springer (2009), https://doi.org/10.1007/978-3-642-00457-5_18
51. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) CCS 2018. pp. 525–537. ACM (2018), https://doi.org/10.1145/3243734.3243805
52. Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 335–354. Springer, Heidelberg (Aug 2004)
53. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part II. LNCS, vol. 10678, pp. 139–171. Springer, Heidelberg (Nov 2017)
54. Khurana, D., Sahai, A.: Two-message non-malleable commitments from standard sub-exponential assumptions. IACR Cryptology ePrint Archive 2017, 291 (2017), http://eprint.iacr.org/2017/291
55. Lin, H., Liu, T., Wee, H.: Information-theoretic 2-round MPC without round collapsing: Adaptive security, and more. In: Pass, R., Pietrzak, K. (eds.) TCC 2020. vol. 12551, pp. 502–531. Springer (2020), https://doi.org/10.1007/978-3-030-64378-2_18
56. Mohassel, P., Rosulek, M.: Non-interactive secure 2PC in the offline/online and batch settings. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 425–455. Springer, Heidelberg (Apr / May 2017)
57. Ostrovsky, R., Richelson, S., Scafuro, A.: Round-optimal black-box two-party computation. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 339–358. Springer, Heidelberg (Aug 2015)
58. Paskin-Cherniavsky, A.: Secure Computation with Minimal Interaction. Ph.D. thesis, Technion (2012), available at http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2012/PHD/PHD-2012-16.pdf
59. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (Feb 2004)
60. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: 51st FOCS. pp. 531–540. IEEE Computer Society Press (Oct 2010)
61. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: 27th FOCS. pp. 162–167. IEEE Computer Society Press (Oct 1986)