Separating Two-Round Secure Computation from Oblivious Transfer

Benny Applebaum
Tel-Aviv University, Israel
bennyap@post.tau.ac.il

Zvika Brakerski
Weizmann Institute of Science, Rehovot, Israel
zvika.brakerski@weizmann.ac.il

Sanjam Garg
UC Berkeley, CA, USA
sanjamg@berkeley.edu

Yuval Ishai
Technion – Israel Institute of Technology, Haifa, Israel
yuvali@cs.technion.ac.il

Akshayaram Srinivasan
UC Berkeley, CA, USA
akshayaram@berkeley.edu

Abstract

We consider the question of minimizing the round complexity of protocols for secure multiparty computation (MPC) with security against an arbitrary number of semi-honest parties. Very recently, Garg and Srinivasan (Eurocrypt 2018) and Benhamouda and Lin (Eurocrypt 2018) constructed such 2-round MPC protocols from minimal assumptions. This was done by showing *a round preserving reduction* to the task of secure *2-party* computation of the oblivious transfer functionality (OT). These constructions made a novel non-black-box use of the underlying OT protocol. The question remained whether this can be done by only making black-box use of 2-round OT. This is of theoretical and potentially also practical value as black-box use of primitives tends to lead to more efficient constructions.

Our main result proves that such a black-box construction is impossible, namely that non-black-box use of OT is necessary. As a corollary, a similar separation holds when starting with any 2-party functionality other than OT.

As a secondary contribution, we prove several additional results that further clarify the landscape of black-box MPC with minimal interaction. In particular, we complement the separation from 2-party functionalities by presenting a complete 4-party functionality, give evidence for the difficulty of ruling out a complete 3-party functionality and for the difficulty of ruling out black-box constructions of 3-round MPC from 2-round OT, and separate a relaxed "non-compact" variant of 2-party *homomorphic secret sharing* from 2-round OT.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Cryptographic protocols

Keywords and phrases Oracle Separation, Oblivious Transfer, Secure Multiparty Computation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.71

Funding *Benny Applebaum*: Supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, and the Check Point Institute for Information Security.
Zvika Brakerski: Supported by the Binational Science Foundation (Grant No. 2016726), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).
Sanjam Garg: Supported in part from AFOSR Award FA9550-19-1-0200, AFOSR YIP Award, NSF CNS Award 1936826, DARPA and SPAWAR under contract N66001-15-C-4065, a Hellman Award

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).
Editor: Thomas Vidick; Article No. 71; pp. 71:1–71:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and research grants by the 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.
Yuval Ishai: Supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, BSF grant 2018393, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India.
Akshayaram Srinivasan: Supported in part from AFOSR Award FA9550-19-1-0200, AFOSR YIP Award, NSF CNS Award 1936826, DARPA and SPAWAR under contract N66001-15-C-4065, a Hellman Award and research grants by the 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.

Acknowledgements We thank Iftach Haitner, Mohammad Mahmoody, and Rotem Tsabary for helpful discussions.

1 Introduction

Secure multiparty computation (MPC) allows mutually distrusting parties to compute a joint function f of their private inputs without revealing anything more than the output to each other.

In this paper we consider the simplest setting for MPC with *no honest majority*, namely MPC with an arbitrary number of corrupted parties. We focus on the *semi-honest* (aka passive) security model, where corrupted parties follow the protocol but try to (jointly) learn additional information on inputs of uncorrupted parties from the messages they observe. We also assume that the parties can communicate over *secure point-to-point channels* and that corruptions are *non-adaptive* (i.e., the set of corrupted parties is fixed before the protocol's execution). All of the above assumptions make negative results stronger.

The design and analysis of MPC protocols crucially rely on the notion of *secure reductions*. In particular, classical completeness results [54, 34] have shown that the problem of securely computing a general n-party functionality f efficiently reduces to the problem of securely computing the elementary finite 2-party *Oblivious Transfer* (OT) functionality [51, 26]. (Similar results have been proven for active adversaries as well [44, 43].) Perhaps surprisingly, for 2-party secure computation (2PC), Yao's reduction is *round preserving*. That is, it incurs no overhead in the round complexity. It additionally requires the parties to make a black-box use of any pseudorandom generator (PRG).

▶ **Theorem 1** (Round-optimal 2PC [54]). *Every 2-party functionality g admits an MPC protocol that only makes parallel calls to an OT oracle and a black-box use of a PRG.*

In more detail, the OT functionality F_{OT} involves two parties referred to as *Receiver* and *Sender*. The functionality takes a bit x from the Receiver and a pair of bits (more generally, strings) (m_0, m_1) from the Sender, and delivers to the Receiver the message m_x. This is done while hiding m_{1-x} from the Receiver and hiding x from the Sender.

Yao's reduction makes a single round of parallel calls to F_{OT}.[1] Using a suitable composition theorem for MPC in the semi-honest model (see, e.g., [20, 33]), this can be securely replaced by parallel invocations of any *OT protocol*, namely a secure 2-party protocol for F_{OT}. The

[1] If both parties should receive an output, the reduction uses parallel OTs in both directions, where each party acts both as a Sender and as a Receiver.

resulting construction of 2-party MPC from a 2-party OT protocol is *black-box*.[2] This means that the MPC protocol does not depend on the code of the underlying OT protocol, and moreover the security proof is black-box in the sense that any adversary "breaking" the MPC protocol can be used as a black-box to break the OT protocol. Instantiating with one of several natural known 2-round OT protocols (whose existence follows from standard intractability assumptions), we get a 2-round 2-party MPC protocol, which is clearly optimal.

Round Complexity in the Multiparty Setting. In contrast to the 2-party setting, progress on the round complexity of general MPC has been slow and some of the questions still remain unanswered. As already mentioned, the completeness of OT in the multiparty setting was first established by Goldreich, Micali, and Wigderson (GMW) [34]. However, their reduction suffered from large round complexity (proportional to the circuit depth of the target function). The question of achieving a constant-round protocol has been considered by Beaver, Micali, and Rogaway [12], who extended Yao's garbled circuit technique to the multiparty setting. Combined with the GMW result, this yields a reduction to OT with constant overhead in the round complexity.

▶ **Theorem 2** (Constant-round MPC [34, 12]). *Every n-party functionality admits a constant-round protocol, making black-box use of a PRG, given an oracle access to F_{OT}.*

In more concrete terms, the most round-efficient current MPC protocol that makes a black-box use of a 2-round OT protocol requires 4 rounds of interaction [1]. The above results left a gap between the round complexity of 2PC and MPC protocols. In a recent breakthrough, this gap was partially closed.

▶ **Theorem 3** (2-round MPC from minimal assumption [31, 13]). *Suppose a 2-round OT protocol exists. Then every n-party functionality admits a 2-round MPC protocol.*

The theorem settles the high-order bit about the minimal assumptions needed for 2-round MPC by showing that 2-round OT is sufficient. (Being a special case of 2-round MPC, it is clearly necessary.) However, quite surprisingly, the MPC protocol in these works inherently makes use of the code of the underlying OT protocol. This situation is quite rare in the context of MPC protocols and in cryptography in general (see Section 1.2), and it is not clear whether this non-black-box use of OT is inherent. This calls for the following natural question:

> Is it possible to reduce general n-party MPC to a 2-party OT protocol in a round-preserving black-box way? In particular, is there a black-box construction of 2-round MPC from 2-round OT?

The above question is not only of a theoretical interest, but is also potentially relevant to practice. Indeed, black-box use of cryptographic primitives tends to lead to more efficient constructions. The goal of obtaining efficient 2-round MPC protocols is very well motivated, since such protocols have *qualitative* advantages over similar protocols with a bigger number of rounds. Indeed, in a 2-round MPC protocol, each party can send its first-round messages and then go offline until all second-round messages are received and the output can be computed. Moreover, the first-round messages can be potentially reused for several computations in which a party's input remains unchanged. This is analogous to the qualitative advantage of public-key encryption over interactive key agreement.

[2] The notion of a black-box construction used in this paper (also referred to as a black-box reduction) corresponds to the notion of a *fully black-box reduction* in the taxonomy of [52].

In this paper, we will provide a negative answer to the above question, showing that there is a real gap between the power of round-preserving black-box (RPBB) reductions and round-preserving non-black-box reductions. Our findings also reveal a rich and somewhat unexplored world of cryptographic protocols that use a minimal amount of interaction. We will exhibit some of the black-box and non-black-box connections among these primitives and relate them to standard ones.

1.1 Our Results

We now give a more detailed account of our results.

1.1.1 Separating 3-Party Functionalities from 2-Party Functionalities

Our first result shows that 2-round MPC protocols cannot be based on 2-round OT in a black-box way. In fact, we show that even 3-party computation of fairly simple functionalities is unachievable via black-box use of 2-round OT.

▶ **Theorem 4** (Main Result). *There exists a 3-party functionality f that cannot be securely computed by a 2-round protocol with black-box use of 2-round OT.*

We note that we do not just rule out round-preserving reductions to the ideal-OT functionality, but rather rule out all such black-box constructions from an *OT protocol.* (Indeed, much of the technical work is devoted to coping with the latter model; see Section 2.) Moreover, the theorem holds even for constructions in the private-channel setting (where each pair of parties is connected via a private channel), and even when the parties have an access to a public common reference string (CRS), and to a random oracle.

1.1.2 A Complete 4-Party Functionality

OT turns out to be incomplete for MPC under RPBB reductions. Given this state of affairs, one may try to prove a completeness result for some other finite functionality. We show that this is indeed possible. Specifically, let $(3,4)-\mathsf{MULTPlus}$ denote the 4-party functionality that takes a pair of bits (x_i, z_i) from each of the first three parties (and no input from the fourth party) and delivers the value $x_1x_2x_3+z_1+z_2+z_3$ to all four parties where addition and multiplication are over the binary field. We prove that $(3,4)-\mathsf{MULTPlus}$ is MPC-complete under RPBB reductions. (Related results have been proved in other settings [18, 28].) In fact, we prove an "ideal-oracle" completeness result just like in Yao's theorem.

▶ **Theorem 5** (Round Optimal MPC from Finite Ideal Functionality). *Every n-party functionality f can be realized using parallel calls to a $(3,4)-\mathsf{MULTPlus}$ oracle and a black-box use of a PRG.*

It's worth noting that $(3,4)-\mathsf{MULTPlus}$ is related to the standard 2-party OT functionality. In general, for $d \leq p$, let $(d,p)-\mathsf{MULTPlus}$ denote the p-party functionality in which each of the first d parties holds an input (x_i, z_i) and the product-sum $\prod_i x_i + \sum_i z_i$ is delivered to all p parties. Then, standard 2-party OT is equivalent (under RPBB reductions) to $(2,2)-\mathsf{MULTPlus}$.[3]

[3] First observe that the receiver's output in OT can be written as $m_0 + x(m_1 - m_0)$ where addition and multiplication are over the binary field. Therefore, we can implement OT based on $(2,2)-\mathsf{MULTPlus}$ by letting the receiver (resp., sender) play the role of the first party (resp., second party) with inputs $x_1 = x$

1.1.3 The Land of Three-Party Functionalities

The finite $(3, 4) - \mathsf{MULTPlus}$ therefore stands at the entry point to the general MPC mainland. Across the ocean, lies the island of two party functionalities (including the complete OT) and one cannot cross it in a black-box round-preserving vessel. We move on and explore the mysterious land of three party functionalities.

Given the incompleteness of 2-party functionalities and the completeness of four party functionalities (under round-preserving BB reductions), it is natural to ask whether 3-party functionalities are complete. We show that the answer to this question is related to a well-known open problem in information-theoretic cryptography.

▶ **Question 6** ([41]). *Does every finite function admit a degree-2 statistical randomized encoding?*

A randomized encoding (RE) of a function [41, 6] $f(x)$ is a randomized function $\hat{f}(x; r)$ that, in addition to the input x, takes an additional random input r. For any input x, the random variable $\hat{f}(x)$, induced by a random choice of r, should reveal the value of $f(x)$ and hide everything else. The power of REs stems from the fact that even complicated functions can be encoded by simple encoding. In the context of MPC, it is known that every finite function can be encoded by a function $\hat{f}(x; r)$ that each of its outputs can be written as a degree-3 polynomial over the indeterminates (x, r). While some negative results are known for perfectly-private degree-2 encodings [41], the feasibility of statistically-private degree-2 encodings (that are allowed to have a small non-zero privacy error) has remained open for almost 20 years. (See also the surveys [40, 3].) We relate this longstanding open problem to the completeness of 3-party functionalities under RPBB reductions.

▶ **Proposition 7.** *A positive answer to Question 6 implies that every n-party finite functionality g can be realized using parallel calls to an oracle that implements the 3-party functionality* $(2, 3) - \mathsf{MULTPlus}$.

The proposition implies that we cannot rule out the completeness of 3-party functionalities without ruling out the existence of general degree-2 (statistical) randomized encoding. Similar barriers have been established in the context of degree-2 cryptographic hash functions [5]. We note that the completeness of $(2, 3) - \mathsf{MULTPlus}$ follows even from the existence of general degree-2 fully-secure *multiparty randomized encoding* [4] – a seemingly weaker variant of RE whose existence is also open. (See the discussion in [4].)

External output functionalities. The $(2, 3) - \mathsf{MULTPlus}$ is a special case of an *external-output* 3-party functionality. Formally, let $g(x, y)$ be a 2-party functionality. The *external version* of g, is the 3-party functionality f_g that takes x from Alice, y from Bob and delivers $g(x, y)$ to Alice and Bob, and to Carol who holds no input.[4] Two-round protocols for

and a random bit z_1 (resp., $x_2 = m_1 - m_0$ and $z_2 = m_0$). The receiver gets the required output (by subtracting z_1 from the output of the $(2, 2) - \mathsf{MULTPlus}$), and the sender learns noting (since it receives a random bit). In the other direction, first observe that the one-sided variant of $(2, 2) - \mathsf{MULTPlus}$, where only the first party has to learn the value $x_1x_2 + z_1 + z_2$, is RPBB-reducible to OT. Indeed, let player 1 (resp., player 2) play the role of the receiver (resp., sender) with inputs $x = x_1$ (resp., $m_0 = z_2$ and $m_1 = z_2 + x_2$), and set the output of player 1 to be the output of the OT plus z_1. Next, observe that standard $(2, 2) - \mathsf{MULTPlus}$ can be constructed by making two parallel calls to the one-sided variant.

[4] In fact, for all of our purposes, an even weaker version suffices. In this relaxed version, all parties are allowed to learn the output (for purposes of privacy), but only Carol is *required* to learn it (for purposes of correctness). Since this leads to a cumbersome definition, we stick to the simpler version described above.

such functionalities turn out to have interesting properties. Specifically, at the core of our main impossibility result (Theorem 4), lies the following constructive theorem for external functionalities.

▶ **Theorem 8** (Conversion Theorem). *Let $g(x, y)$ be a 2-party functionality. The external version of g is the 3-party functionality f_g that takes x from Alice, y from Bob and delivers $g(x, y)$ to Alice and Bob, and to Carol who holds no input.*

Suppose that the functionality f_g can be securely computed in 2 rounds by making a black-box use of 2-round OT over private channels. Then, f_g can be securely implemented over random inputs given only an access to a Random Oracle over private channels. Moreover, in the resulting protocol Carol sends no message and so it yields a two-party protocol for computing g over random inputs given only an access to a Random Oracle.

Haitner et al. [36] showed that any 2-party functionality that can be securely realized in the Random Oracle (RO) model is *trivial* in the sense that it admits an unconditional 2-party protocol over random inputs with security against computationally-unbounded adversaries. As a corollary, we derive the following stronger version of Theorem 4.

▶ **Corollary 9.** *Every external functionality f_g that is based on a non-trivial 2-party functionality f cannot be computed by a 2-round protocol that makes a black-box use of 2-round OT even in the private-channel setting.*

A notable example for such a non-trivial 2-party functionality is the AND functionality [21, 46].

Corollary 9 is tight in terms of round complexity. With one additional round, f_g can be black-box reduced to 2-round OT. (Specifically, one can use Yao's theorem to pass the value of $f(x, y)$ to Alice and Bob in two rounds, and then exploit the additional round to send this value to Carol.) The 2-party completeness of OT (Theorem 1) also implies that Corollary 9 holds when the OT functionality is replaced by an arbitrary 2-party functionality $h(x, y)$. Overall, we get a separation between all 2-party functionalities and all external functionalities f_g whose underlying g is non-trivial.

Relation with homomorphic secret sharing. Two-round MPC for extended-output functionalities can be seen as closely related to the problem of homomorphic secret sharing (HSS) [16, 18]. HSS is the secret-sharing analogue of fully homomorphic encryption. A (2-party) HSS scheme allows local computation of a function $g(x, y)$ on independently shared inputs x and y, where the output $g(x, y)$ can be decoded from the pair of output shares. The standard notions of HSS require either *additive* decoding over a group or, more generally, that the output shares be *compact* in the sense that they are shorter than the inputs. A natural variant is to replace compactness by the requirement that the pair of output shares give no information except $g(x, y)$, even from the point of view of one of the input holders. This security requirement is easily obtained from additive HSS via a simple additive refreshing of the output shares. This flavor of non-compact HSS easily implies (in a black-box way) a 2-round external output protocol for g (in the private-channel setting), which by Corollary 9 can be separated from 2-round OT. On the other hand, a non-black-box construction of non-compact HSS from 2-round OT follows from [31, 28].

1.2 Discussion

In this section we give some further perspective on our results and some future research directions which they motivate.

1.2.1 Why is the multiparty setting different than the 2-party setting?

It is instructive to reconsider the round complexity of MPC in light of our results. Protocols with low round complexity are based on two types of reductions.

1. A *degree reduction* that takes a general n-party functionality f and reduces it (via RPBB reduction) to a degree-d functionality for a constant d. Specifically, the standard machinery of randomized-encoding leads to degree 3. In the special case of two parties, we can trivially reduce the degree down to 2, and so we get a degree reduction to $d = \min(3, n)$.
2. A *player reduction* that takes an n-party functionality of degree d and reduces it to the (d, p)–MULTPlus functionality. We show that p can be dropped down to $d + 1$ and, in any case, it is no larger than n, leading to an expression of the form $p = \min(d + 1, n)$.

In the special case of two parties $n = 2$ we get an RPBB-reduction to $(2, 2)$–MULTPlus which is equivalent to OT. For large n's, this leads to the completeness of $(3, 4)$–MULTPlus (Theorem 5). In order to prove that OT is complete (under RPBB-reductions) we have to bypass two barriers: A *Degree barrier* (prove completeness of degree 2 functionalities) and a *Player reduction barrier* (reducing the 3-party functionality (2,3)–MULTPlus to (2,2)–MULTPlus). While the first barrier is well-known, the second one appears to be new to this work. Clearly, both barriers are bypassed by non-black-box techniques (Theorem 3). We show that this is inherent for the second "player reduction" barrier, and leave the possibility of breaking the degree-barrier via RPBB-reduction open.

1.2.2 On the Role of Non-Black-Box Constructions in Cryptography

Our main result provides a very natural example of a pair of cryptographic primitives for which a non-black-box construction of one from the other exists but a black-box construction can be ruled out. Thus, our work further demonstrates the essential role of non-black-box techniques in cryptography.

To give some historical perspective, following the seminal result of Impagliazzo and Rudich [39] and subsequent works on black-box separations in cryptography [53, 32, 52], the question of finding a pair of "natural" cryptographic primitives for which a non-black-box reduction is provably necessary has been put forward as a desirable but elusive goal.[5] For some of the conjectured candidate examples, such as constructing "malicious OT" from "semi-honest OT," black-box constructions were subsequently found [35]. However, in recent years several such provable examples emerged. We survey some of the most notable ones below.

- *Non-interactive commitments from OWFs*: Mahmoody and Pass [48] showed that non-interactive commitments cannot be constructed from so-called "hitting-OWFs" in a black-box manner, even though a non-black-box construction was previously shown [10]. One nice feature of this example is that a non-interactive commitment is a very basic primitive. However, in comparison hitting-OWFs have found little other applications in cryptography. Furthermore, the separation here is intuitively weak since knowing the *circuit size* of the OWF enables a black-box construction. This is contrasted with the non-black-box constructions of 2-round MPC from OT [31, 13], which make an essential use of the full code of the OT protocol.

[5] The question is informal due to the subjective nature of the term "natural primitive." It should not be confused with the question of black-box vs. non-black-box *simulation*, for which Barak's breakthrough non-black-box simulation technique [8] gave the first such natural examples.

- *Two-round OT extension*: Beaver gave a construction of two-round OT extension [11] making a non-black-box use of one-way functions. This construction can be cast in the OT-hybrid model. However, very recently, Garg et al. [29] showed that a back-box variant of such a constriction is impossible. They showed that such constructions are not possible even when black-box use of a random oracle (and not just a one-way function) is allowed. One limitation of this example is that the separation is only proved for protocols in the OT-hybrid model.
- *IBE from CDH (or Generic Groups)*: In a recent result, Döttling and Garg [24] show that Identity-Based Encryption (IBE) can be realized under the Computational Diffie-Hellman (CDH) assumption, while black-box constructions of the same had been previously ruled out [15, 50]. However, in this case both the positive and the negative result use strong "structured" primitives.

  In another very related example, Döttling and Garg [23] showed a generic non-black-box construction of hierarchical-IBE from IBE but we can expect a black-box impossibility for the same using techniques from [15].[6]
- *Constructions of IO*: In a very recent work, Garg et al. [30] showed that indistinguishability obfuscation (IO) [27] cannot be constructed from compact functional encryption (FE) in a black-box manner, even though non-black-constructions achieving this were already known [14, 2].
- *Secret-Key FE vs Public-Key FE*: In a recent work, Kitagawa et al. [45] showed that public-key FE can be constructed from secret-key FE in a non-black-box manner, even though black-box positive constructions had been previously ruled out [7].

In comparison with the above works, our main result has the advantage that it considers two very natural and simple primitives. Our separation lives entirely in the "passive adversary" world, and does not depend on the input domain being super-polynomial. For instance, our separation is also meaningful for MPC with a uniform input distribution over a constant-size domain. Thus, it is arguably similar in spirit to the Impagliazzo-Rudich separation of key agreement from one-way functions [39], except that in the latter case no analogous non-black-box construction is known.

1.3 Open Problems

While we settle the main open question concerning black-box round-optimal MPC, our work leaves several interesting directions for future research. We highlight a few below, focusing on our current setting of semi-honest security with no honest majority.

- **3-round MPC from black-box OT.** Our main result rules out 2-round MPC protocols making a black-box use of 2-round OT. On the other hand, a previous result of Ananth et al. [1] shows that such 4-round protocols exist. What about 3-round protocols? In the full version of the paper, we give evidence that extending our negative result to the 3-round case would require settling Question 6 in the negative. This barrier does not seem to apply to 3-round protocols in which the first round messages do not depend on the inputs, or alternatively 2-round protocols with a public-key infrastructure (PKI) setup.
- **Black-box use of stronger primitives.** Can our negative result be bypassed by replacing OT with stronger or more structured primitives? It is known that 2-round MPC can make black-box use of different flavors of multi-key homomorphic encryption [49, 22]

[6] Even though we expect such an impossibility to hold, we are not aware of a work that gives a full proof of this claim.

or homomorphic secret sharing [17, 18]. However, this is almost immediate from the definitions of such primitives. Using simpler structured primitives, such as a "DDH-hard" group or a generic group, we have black-box 2-round protocols that require a PKI setup [28]. Can we get similar group-based constructions in the plain model? Alternatively, can the separation be bypassed by using stronger variants of 2-round OT, such as OT with high information rate [25] or OT with a stronger notion of receiver privacy [42]?

- **Minimal complete primitive for 2-round MPC.** We have shown the existence of a 4-party functionality such that general MPC reduces to *parallel* calls to this functionality without further interaction. We have also ruled out such a 2-party functionality. This leaves open the 3-party case. As in the case of 3-round MPC from 2-round OT, we can show that proving a *negative* result would require settling Question 6 in the negative.
- **Standard MPC vs. client-server MPC.** Our main negative result automatically carries over to the stronger *client-server model* for MPC, where n clients interact with n servers who have no inputs or outputs. It is known that 2-round client-server MPC can be constructed in a non-black-box way from standard 2-round MPC [28]. Whether such a black-box construction exists remains open.

2 Technical Overview

In this section, we give a high-level overview of our techniques in proving the main result (Theorem 4). To keep the exposition simple, we restrict ourselves to proving the impossibility result for securely computing external-AND.

External-AND Functionality. Let us denote the three parties by (P_1, P_2, P_3). The private input of P_1 is a bit x, the private input of P_2 is a bit y and P_3 does not have any private inputs. The functionality $f_\times$ outputs $x \cdot y$ to all the parties. Specifically, $f_\times(x, y, \perp) = x \cdot y$.

Main Idea. To prove the impossibility result, we define a set of oracles such that 2-round oblivious transfer exists with respect to these oracles, but there exists no 2-round, semi-honest protocol for securely computing $f_\times$. This is sufficient to rule out a black-box transformation from 2-round oblivious transfer to 2-round, 3-party semi-honest protocols for general functionalities. Below, we describe these oracles (throughout this overview, sec denotes the security parameter):

- OT_1 is a random length tripling function that takes in the receiver's choice bit $b \in \{0,1\}$ and its random tape $r \in \{0,1\}^{\text{sec}}$ and outputs the receiver's message otm_1.
- OT_2 is a random length tripling function that takes in the receiver's message otm_1, the sender's inputs $m_0, m_1 \in \{0,1\}$, its random tape $s \in \{0,1\}^{\text{sec}}$ and outputs the sender's message otm_2.
- OT_3 is a function that takes the transcript $(\mathsf{otm}_1, \mathsf{otm}_2)$ along with (b, r) as input and outputs m_b if there exists unique (m_0, m_1, s) for which $\mathsf{OT}_1(b, r) = \mathsf{otm}_1$ and $\mathsf{OT}_2(\mathsf{otm}_1, m_0, m_1, s) = \mathsf{otm}_2$. Otherwise, it outputs $\perp$.

As observed by [37], the oracles $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ naturally give rise to a 2-round oblivious transfer protocol. Specifically, letting b, r denote the input/randomness of the receiver, and letting $(m_0, m_1), s$ denote the input/randomness of the sender, the protocol proceeds as follows: The receiver sends $\mathsf{otm}_1 = \mathsf{OT}_1(b, r)$ to the sender, who responds with $\mathsf{otm}_2 = \mathsf{OT}_2(\mathsf{otm}_1, m_0, m_1, s)$, allowing the receiver to output the value $\mathsf{OT}_3(\mathsf{otm}_1, \mathsf{otm}_2, b, r)$.

In this work, we prove that the existence of a 2-round protocol for external-AND w.r.t. $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ implies a *two-party* protocol for computing $g(x, y) = x \cdot y$ in the random oracle model. (Note that we start with a three-party protocol for an external functionality, and show a two-party protocol for a related functionality.) The existence of such two-party protocol is known to be impossible [21, 46, 36, 47] and therefore the original protocol can also not exist. This proves Theorem 8 discussed above, and implies Theorem 4 as a corollary.

Outline. The above result is proven using a sequence of transformations depicted in Figure 1.

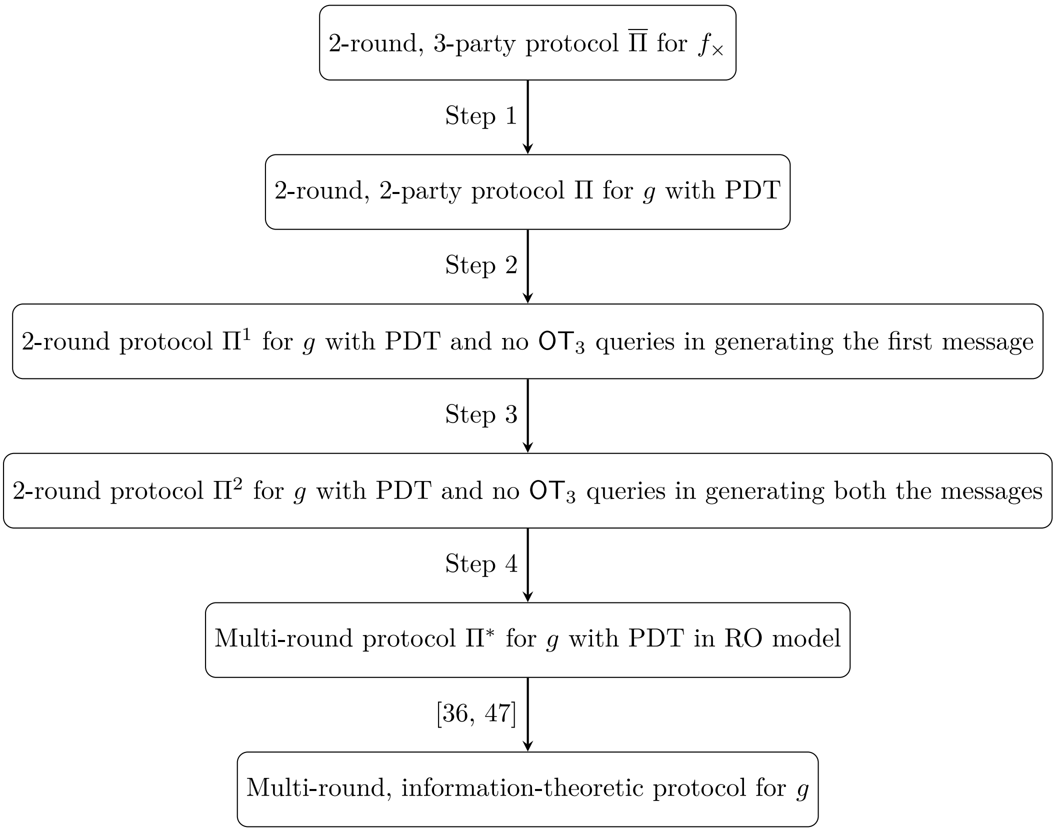
2-round, 3-party protocol $\overline{\Pi}$ for $f_\times$
Step 1
2-round, 2-party protocol Π for g with PDT
Step 2
2-round protocol Π^1 for g with PDT and no OT_3 queries in generating the first message
Step 3
2-round protocol Π^2 for g with PDT and no OT_3 queries in generating both the messages
Step 4
Multi-round protocol Π^* for g with PDT in RO model
[36, 47]
Multi-round, information-theoretic protocol for g

Figure 1 Key Steps in the Proof. Here, PDT denotes publicly decodable transcript, $g(x, y) = x \cdot y$ and $f_\times(x, y, \perp) = g(x, y)$.

Step-1: Publicly Decodable Transcript. Let $\overline{\Pi}$ be a 2-round protocol for securely computing $f_\times$ w.r.t. $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$. We first show that this implies a 2-round, 2-party protocol Π for computing the two-party functionality $g = g(x, y) = x \cdot y$, which has an additional special property – the output is *publicly decodable* from the transcript. More formally, there exists a deterministic algorithm Dec that computes the output of the functionality given the transcript of the two-party protocol. In particular, if there exists a protocol Π that computes g with publicly decodable transcript, then Dec on input $\mathbb{T}$ (which is the transcript of the protocol Π) outputs $g(x, y)$. In terms of security, Π is required to have the standard security properties of a two-party (semi-honest) protocol, i.e., the corrupted party does not learn any information about the other party's input except the output.

To transform a 2-round protocol for $f_\times$ into a 2-round protocol Π for g with publicly decodable transcript, we use a player emulation technique.[7] Concretely, we ask P_1 to choose a uniform random tape for P_3 and send this random tape in the first round. Using this random

[7] The idea of player emulation goes back to [19]; see also [38].

tape, P_1 and P_2 can generate the messages P_3 would have sent in the original protocol. Additionally, P_1 and P_2 forwards all its outgoing messages that are sent to each other as well the messages sent to P_3 in the original protocol.

This protocol satisfies public decodability since given the transcript of the protocol (which includes the entire view of P_3 in the original protocol), one can run the output computing algorithm of P_3 to learn $g(x, y)$. Further, the security follows directly from the security of the original protocol when (P_1, P_3) and (P_2, P_3) are corrupted.[8]

Remaining Steps – Removing OT_3 Queries. In the remainder of the proof, we show that use of the oracle OT_3 can be removed. More specifically, we show how to convert any two-round two-party secure computation protocol Π with access to $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ and publicly decodable transcript into a two-party protocol that computes the same functionality, but with a few differences. The oracles OT_3 will no longer be used in the new protocol, but this will come at a cost, both in round-complexity and in security:

- The round complexity of the protocol will grow by a polynomial factor (essentially upper bounded by the query complexity of Dec).
- The correctness and security guarantees will only be with respect to random inputs (we call this "weak security"). One instructive way to think about weak security is to think of a protocol between parties that have no input, and at the beginning of the execution they sample a random input using their local random tape (or shared randomness) and proceed to execute the protocol. Note that this makes simulation easier since we no longer need to worry about consistency with an adversarially chosen (or sampled) input.

In other words, the new protocol Π^* only makes queries to $(\mathsf{OT}_1, \mathsf{OT}_2)$ which are essentially random oracles. Therefore, Π^* securely computes g in the random oracle model. However, it follows from [36, 47] that such a protocol can be used to securely compute g in the information-theoretic setting and this is known to be impossible for the AND functionality [21, 46] (even with weak security as described above).

The remainder of the overview describes this transformation. We transform Π to Π^* through a sequence of steps. We first transform Π to Π^1 in which the first message function of the protocol does not make any OT_3 queries (Step 2 below). Then, we transform Π^1 to Π^2 such that the first and second message functions of the protocol do not make any OT_3 queries (Step 3 below). Finally, we transform Π^2 to Π^* such that the decoder Dec does not make any OT_3 queries (Step 4 below). It is the final step that incurs the blow-up in the round complexity. Additional details follow.

Step-2: $\Pi \Rightarrow \Pi^1$. The first message function of Π has access to $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ oracles and may make multiple queries to all of them. In order to perform this transformation, we devise a mechanism to emulate the OT_3 oracle without making actual queries to it. Recall that any query to the OT_3 oracle contains $((\mathsf{otm}_1, \mathsf{otm}_2), (b, r))$ and it outputs m_b if and only if there exists (m_0, m_1, s) for which $\mathsf{OT}_1(b, r) = \mathsf{otm}_1$ and $\mathsf{OT}_2(\mathsf{otm}_1, m_0, m_1, s) = \mathsf{otm}_2$. The first step of the OT_3 oracle is easy to emulate; we can query OT_1 on (b, r) and check if the output is otm_1. To emulate the second step, we maintain a list of all the queries/responses made by the first message function to OT_2. If we find an entry $(\mathsf{otm}_1, m_0, m_1, s, \mathsf{otm}_2)$ in this list, we output m_b; else, we output $\perp$. Note that since OT_2 is length tripling, it is

[8] It is easy to see that the security of the transformed protocol requires security against collusion of P_1, P_3 since P_1 has the entire view of P_3. We also require security against (P_2, P_3) collusion since P_1 forwards all its messages sent to P_3 in the second-round of the protocol.

injective with overwhelming probability. Thus, if we find such an entry then our emulation is correct. On the other hand, if we don't find such an entry, we output $\perp$ and it can be easily shown that the original oracle also outputs $\perp$ except with negligible probability. Thus, our emulation is statistically close to the real oracle.

Step-3: $\mathbf{\Pi^1 \Rightarrow \Pi^2}$. It might be tempting to conclude that a similar strategy as before should work even for OT_3 queries made in the second round. That is, maintain the list of queries to the OT_2 oracle and when the second message function makes an OT_3 query, check if there is entry in this list with the response equal to otm_2. If such an entry is found, output the corresponding m_b; else, output $\perp$. This strategy fails because in the second round it is possible that the relevant OT_2 query was made by the other party and therefore it is not possible for each party to only consider the list of OT_2 queries made locally. Note, however, that only one simultanous round of communication has been made by the parties so far. Therefore, it must be the case that the party that made the OT_2 query also made the respective OT_1 query.

To take care of such queries, that we call "correlated queries", we modify the first round of Π^1 as follows. The parties will prepare an additional list L that contains all correlated queries that are "likely" to be asked by the other party. (No OT_3 calls will be made while preparing this list.)

The parties will now send this list L along with the first round message of Π^1. Now, when the second message function of a party in Π^1 attempts to make an OT_3 query on $(\mathsf{otm}_1, \mathsf{otm}_2, (b, r))$, we first check if otm_1 is valid (by querying OT_1) and then answer this query as follows. If otm_2 is a result of a local query then find the response using the list of local queries/responses. If otm_2 is a correlated query, use the list L sent by the other party to answer. If we don't find any entry in the local list or the correlated list, we output $\perp$. We show that with overwhelming probability, the real oracle also outputs $\perp$ in this case. We also prove that sending this additional list of "likely" correlated queries does not harm the security of Π^2.

To conclude, we describe how the list L is generated, say by P_1. Note that the list needs to be generated at a point where P_1 already decided on its first Π^1 message; now it just needs to come up with L. To this end, P_1 executes many copies of Π^1 executions of P_2, each time with fresh randomness and random input. Then the list L contains the responses to the list of all correlated OT_2 queries, i.e., the valid queries made to OT_3 by "virtual" P_2 such that both OT_1 and OT_2 have been generated by P_1. This will allow to preserve correctness on an average input, and does not violate privacy since given the first Π^1 messages, anyone can sample such executions.

Step-4: $\mathbf{\Pi^2 \Rightarrow \Pi^*}$. At the end of step-3, we have a protocol where the first and the second message functions do not make any queries to the OT_3 oracle. However, for the parties to learn the output, they must run the decoder Dec on the transcript, and this decoder might make queries to OT_3. Recall that Dec is a deterministic decoding function whose input is the transcript of the interaction. Further recall that Π^* will be a protocol that does not use OT_3 but will have many communication rounds.

In Π^*, the parties will first execute the two rounds of Π^2 to obtain a transcript. Then they will jointly execute the decoder, where for each OT_3 query that the decoder needs to make, the party that made the relevant OT_2 query will "help out" by sending the decoding value to the other party. This will proceed for as many rounds as the number of queries that Dec needs to make, but eventually it will allow both parties to complete the execution of Dec locally and compute the output of the functionality. We will then need to show that privacy is not harmed in this process. Details follow.

Let us go back to the point where both parties finished executing the two rounds of Π^2 now wish to engage in joint decoding. One of the parties, say P_1, starts running the decoder on the transcript, and along the way maintain the list of $\mathsf{OT}_1, \mathsf{OT}_2$ made in this process. When the decoder attempts to make an OT_3 query on input $((\mathsf{otm}_1, \mathsf{otm}_2), (b, r))$, P_1 checks if otm_1 is valid (by making a query to OT_1). It then checks if there is an entry $(\mathsf{otm}_1, m_0, m_1, s, \mathsf{otm}_2)$ in the list of OT_2 queries made by the decoder and in the case such an entry is found, it answers with m_b. If such an entry is not found, P_1 checks its local list of queries/responses made to OT_2 during the generation of the first two messages. If it finds an entry $(\mathsf{otm}_1, m_0, m_1, s, \mathsf{otm}_2)$ in that list, it answers with m_b. If even this list does not contain an entry, there are 3 possibilities.

1. otm_2 is not in the image of OT_2 oracle in which case P_1 has to output $\perp$.
2. otm_2 is in the image of OT_2 oracle and P_2 has made this query.
3. otm_2 is in the image of OT_2 oracle and P_2 has not made this query.

The probability that case-3 happens can be shown to be negligible for similar reasons to ones discussed above: if neither party made the relevant OT_2 query then the value otm_2 is almost surely invalid. Thus, P_1 must decide whether it is in case-1 or case-2 and if it is in case-2, it must give the corresponding m_b. To accomplish this, P_1 sends a message to P_2 with (b, otm_2) and asks P_2 to see if there is an entry of the form $(\mathsf{otm}_1, m_0, m_1, s, \mathsf{otm}_2)$ in its local list of queries to OT_2 oracle. If yes, P_2 responds with m_b; else, it responds with $\perp$. P_1 just gives P_2's message as the corresponding response to that query. This blows up the number of rounds of the protocol Π^* proportional to the number of queries made by the decoder.

Observe that Π^* does not make any queries to the OT_3 oracle. At the end, P_1 learns the output $g(x, y)$ and it can send this as the last round message to P_2. Thus, Π^* also has publicly decodable transcript. The correctness of this transformation directly follows since we prove that case-3 happens with negligible probability and if OT_2 is injective (which occurs with overwhelming probability), it follows that if an entry is found in either of the lists of the two parties or on the local list of the decoder, the response given by the emulation is correct.

To see why this transformation is secure, notice that the query $((\mathsf{otm}_1, \mathsf{otm}_2), (b, r))$ is made by the Dec by just looking at the transcript. Hence, there is no harm in P_1 sending (b, otm_2) to the other party. Similarly, if P_2 has indeed made a query to OT_2 such that the response obtained is otm_2, it should follow from the security of Π^2 that the P_2's privacy is not affected if it sends m_b to P_1. Indeed, this information is efficiently learnable given the transcript and an access to the OT_3 oracle. However, there is a subtle issue with this argument which we elaborate next.

Problem of Intersecting Queries. A subtle issue arises when we try to formally reduce the security of Π^* to the security of Π^2. To illustrate this, let us assume the case where P_2 is corrupted. To get a reduction to the security of Π^2, we must give an algorithm that takes the view of P_2 in Π^2 and efficiently generates the view of P_2 in Π^*. In particular, it must generate the additional messages in Π^* given only the view of P_2 in Π^2. This algorithm is allowed to make OT_3 queries as we are trying to give a reduction to the security of Π^2. For the sake of illustration, assume that the Dec makes a single OT_3 query. A natural approach for this algorithm is to take the transcript available in the view of P_2 and start running the decoder on the transcript. When the decoder makes an OT_3 query, the algorithm uses the real OT_3 oracle to respond to this query. However, notice that the algorithm must generate the messages that correspond to answering this OT_3 query in Π^*. Recall that in Π^*, P_1 first checks in its local list whether there an entry of the form $(\mathsf{otm}_1, m_0, m_1, s, \mathsf{otm}_2)$, and only if such an entry is not found, P_1 sends the message (b, otm_2) to P_2. Thus, to generate the transcript of Π^*, the algorithm must somehow decide whether P_1 would find this entry in its local list or not. However, the algorithm is only given the view of P_2 and does not have any information about the queries that P_1 has made to OT_2.

We see that the problem arises when there is an OT_2 query that potentially was made by both parties. To handle this issue, we resort to the notion of *intersection queries* taken from the key-agreement impossibility result [39, 9]. These works show that it is possible, in polynomial time, to recover a superset of all oracle queries made by both parties (with all but small probability). Given this algorithm, we modify the transformation as follows. The parties will first run the two rounds of communication of Π^2. Then they will run the intersection query finder to recover the intersection query superset. We assume for the purpose of this outline this process is deterministic.[9] Now, upon each potential OT_3 query of the decoder, P_1 will look for the preimage query not only in its query history, but also in the superset of intersection queries, and send a message to P_2 only if the preimage is found in either of this. In particular this means that if the preimage is in the intersection query superset, then we are guaranteed that P_1 will not send a message.

The above modified protocol can be efficiently simulated, since the simulator can also run the intersection query finder and recover the same superset as the parties. Now, if OT_3 gives a valid answer, the simulator looks for a preimage in the intersection query superset. If it finds one, then it concludes that P_1 will not send any message to P_2. If not, then it knows that (except with small probability) exactly one of the parties made the preimage query, and it furthermore knows the internal state of one of the parties, so it knows whether this party made the preimage query. This allows the simulator to always deduce which is the party that made the preimage query and simulate appropriately.

References

1 Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A New Approach to Round-Optimal Secure Multiparty Computation. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 468–499. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63688-7_16.

2 Prabhanjan Ananth and Abhishek Jain. Indistinguishability Obfuscation from Compact Functional Encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 308–326. Springer, Heidelberg, August 2015. doi:10.1007/978-3-662-47989-6_15.

3 Benny Applebaum. Garbled Circuits as Randomized Encodings of Functions: a Primer. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography.*, pages 1–44. Springer International Publishing, 2017. doi:10.1007/978-3-319-57048-8_1.

4 Benny Applebaum, Zvika Brakerski, and Rotem Tsabary. Perfect Secure Computation in Two Rounds. In *TCC 2018: 16th Theory of Cryptography Conference, Part I*, Lecture Notes in Computer Science, pages 152–174. Springer, Heidelberg, March 2018. doi:10.1007/978-3-030-03807-6_6.

5 Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-Complexity Cryptographic Hash Functions. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 7:1–7:31. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.7.

[9] We require that the intersection query finder can be ran consistently by various parties, so if it is randomized it suffices that one of the parties sends a random string that will be used as random tape by all relevant parties.

6 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th Annual Symposium on Foundations of Computer Science*, pages 166–175. IEEE Computer Society Press, October 2004. doi:10.1109/FOCS.2004.20.

7 Gilad Asharov and Gil Segev. Limits on the Power of Indistinguishability Obfuscation and Functional Encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 191–209. IEEE Computer Society Press, October 2015. doi:10.1109/FOCS.2015.21.

8 Boaz Barak. How to Go Beyond the Black-Box Simulation Barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115. IEEE Computer Society Press, October 2001. doi:10.1109/SFCS.2001.959885.

9 Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390. Springer, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_22.

10 Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in Cryptography. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315. Springer, Heidelberg, August 2003. doi:10.1007/978-3-540-45146-4_18.

11 Donald Beaver. Correlated Pseudorandomness and the Complexity of Private Computations. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 479–488, 1996. doi:10.1145/237814.237996.

12 Donald Beaver, Silvio Micali, and Phillip Rogaway. The Round Complexity of Secure Protocols (Extended Abstract). In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513. ACM Press, May 1990. doi:10.1145/100216.100287.

13 Fabrice Benhamouda and Huijia Lin. k-Round Multiparty Computation from k-Round Oblivious Transfer via Garbled Interactive Circuits. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 500–532, 2018. doi:10.1007/978-3-319-78375-8_17.

14 Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability Obfuscation from Functional Encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 171–190. IEEE Computer Society Press, October 2015. doi:10.1109/FOCS.2015.20.

15 Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the Impossibility of Basing Identity Based Encryption on Trapdoor Permutations. In *49th Annual Symposium on Foundations of Computer Science*, pages 283–292. IEEE Computer Society Press, October 2008. doi:10.1109/FOCS.2008.67.

16 Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the Circuit Size Barrier for Secure Computation Under DDH. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 509–539. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_19.

17 Elette Boyle, Niv Gilboa, and Yuval Ishai. Group-Based Secure Computation: Optimizing Rounds, Communication, and Computation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 163–193. Springer, Heidelberg, April/May 2017. doi:10.1007/978-3-319-56614-6_6.

18 Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of Homomorphic Secret Sharing. In Anna R. Karlin, editor, *ITCS 2018: 9th Innovations in Theoretical Computer Science Conference*, volume 94, pages 21:1–21:21. LIPIcs, January 2018. doi:10.4230/LIPIcs.ITCS.2018.21.

19 Gabriel Bracha. An O(log n) expected rounds randomized byzantine generals protocol. *J. ACM*, 34(4):910–920, 1987. doi:10.1145/31846.42229.

20 Ran Canetti. Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology*, 13(1):143–202, January 2000. doi:10.1007/s001459910006.

21 Benny Chor and Eyal Kushilevitz. A Zero-One Law for Boolean Privacy (extended abstract). In *21st Annual ACM Symposium on Theory of Computing*, pages 62–72. ACM Press, May 1989. doi:10.1145/73007.73013.

22 Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky Encryption and its Applications. *IACR Cryptology ePrint Archive*, 2016:272, 2016. URL: http://eprint.iacr.org/2016/272.

23 Nico Döttling and Sanjam Garg. From Selective IBE to Full IBE and Selective HIBE. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 372–408. Springer, Heidelberg, November 2017. doi:10.1007/978-3-319-70500-2_13.

24 Nico Döttling and Sanjam Garg. Identity-Based Encryption from the Diffie-Hellman Assumption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 537–569. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63688-7_18.

25 Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor Hash Functions and Their Applications. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, pages 3–32, 2019. doi:10.1007/978-3-030-26954-8_1.

26 Shimon Even, Oded Goldreich, and Abraham Lempel. A Randomized Protocol for Signing Contracts. *Commun. ACM*, 28(6):637–647, 1985. doi:10.1145/3812.3818.

27 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49. IEEE Computer Society Press, October 2013. doi:10.1109/FOCS.2013.13.

28 Sanjam Garg, Yuval Ishai, and Akshayaram Srinivasan. Two-Round MPC: Information-Theoretic and Black-Box. In *TCC 2018: 16th Theory of Cryptography Conference, Part I*, Lecture Notes in Computer Science, pages 123–151. Springer, Heidelberg, March 2018. doi:10.1007/978-3-030-03807-6_5.

29 Sanjam Garg, Mohammad Mahmoody, Daniel Masny, and Izaak Meckler. On the Round Complexity of OT Extension. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 545–574. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96878-0_19.

30 Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. When Does Functional Encryption Imply Obfuscation? In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 82–115. Springer, Heidelberg, November 2017. doi:10.1007/978-3-319-70500-2_4.

31 Sanjam Garg and Akshayaram Srinivasan. Two-Round Multiparty Secure Computation from Minimal Assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 468–499. Springer, Heidelberg, April/May 2018. doi:10.1007/978-3-319-78375-8_16.

32 Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The Relationship between Public Key Encryption and Oblivious Transfer. In *FOCS 2000*, pages 325–335, 2000.

33 Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004. doi:10.1017/CBO9780511721656.

34 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229. ACM Press, May 1987. doi:10.1145/28395.28420.

35 Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-Box Constructions of Protocols for Secure Computation. *SIAM J. Comput.*, 40(2):225–266, 2011. doi:10.1137/100790537.

36 Iftach Haitner, Eran Omri, and Hila Zarosim. Limits on the Usefulness of Random Oracles. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 437–456. Springer, Heidelberg, March 2013. doi:10.1007/978-3-642-36594-2_25.

37 Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On Robust Combiners for Oblivious Transfer and Other Primitives. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 96–113. Springer, Heidelberg, May 2005. doi:10.1007/11426639_6.

38 Martin Hirt and Ueli M. Maurer. Player Simulation and General Adversary Structures in Perfect Multiparty Computation. *J. Cryptology*, 13(1):31–60, 2000. doi:10.1007/s001459910003.

39 Russell Impagliazzo and Steven Rudich. Limits on the Provable Consequences of One-Way Permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61. ACM Press, May 1989. doi:10.1145/73007.73012.

40 Yuval Ishai. Randomization Techniques for Secure Computation. In Manoj Prabhakaran and Amit Sahai, editors, *Secure Multi-Party Computation*, volume 10 of *Cryptology and Information Security Series*, pages 222–248. IOS Press, 2013. doi:10.3233/978-1-61499-169-4-222.

41 Yuval Ishai and Eyal Kushilevitz. Randomizing Polynomials: A New Representation with Applications to Round-Efficient Secure Computation. In *41st Annual Symposium on Foundations of Computer Science*, pages 294–304. IEEE Computer Society Press, November 2000. doi:10.1109/SFCS.2000.892118.

42 Yuval Ishai and Anat Paskin. Evaluating Branching Programs on Encrypted Data. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pages 575–594, 2007. doi:10.1007/978-3-540-70936-7_31.

43 Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding Cryptography on Oblivious Transfer - Efficiently. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, Heidelberg, August 2008. doi:10.1007/978-3-540-85174-5_32.

44 Joe Kilian. Founding Cryptography on Oblivious Transfer. In *20th Annual ACM Symposium on Theory of Computing*, pages 20–31. ACM Press, May 1988. doi:10.1145/62212.62215.

45 Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Obfustopia Built on Secret-Key Functional Encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 603–648. Springer, Heidelberg, April/May 2018. doi:10.1007/978-3-319-78375-8_20.

46 Eyal Kushilevitz. Privacy and Communication Complexity. In *30th Annual Symposium on Foundations of Computer Science*, pages 416–421. IEEE Computer Society Press, October-/November 1989. doi:10.1109/SFCS.1989.63512.

47 Mohammad Mahmoody, Hemanta K. Maji, and Manoj Prabhakaran. Limits of random oracles in secure computation. In Moni Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 23–34. Association for Computing Machinery, January 2014. doi:10.1145/2554797.2554801.

48 Mohammad Mahmoody and Rafael Pass. The Curious Case of Non-Interactive Commitments - On the Power of Black-Box vs. Non-Black-Box Use of Primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 701–718. Springer, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_41.

49 Pratyay Mukherjee and Daniel Wichs. Two Round Multiparty Computation via Multi-key FHE. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 735–763, 2016. doi:10.1007/978-3-662-49896-5_26.

50 Periklis A. Papakonstantinou, Charles W. Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? Cryptology ePrint Archive, Report 2012/653, 2012. URL: https://eprint.iacr.org/2012/653.

51 M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

52 Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of Reducibility between Cryptographic Primitives. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, Heidelberg, February 2004. doi:10.1007/978-3-540-24638-1_1.

53 Daniel R. Simon. Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, Heidelberg, May/June 1998. doi:10.1007/BFb0054137.

54 Andrew Chi-Chih Yao. How to Generate and Exchange Secrets (Extended Abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, October 1986. doi:10.1109/SFCS.1986.25.