The Price of Low Communication in Secure Multi-party Computation

Juan Garay[1(✉)], Yuval Ishai[2], Rafail Ostrovsky[3], and Vassilis Zikas[4]

[1] Yahoo Research, Sunnyvale, USA
juan.a.garay@gmail.com
[2] Department of Computer Science, Technion and UCLA, Haifa, Israel
yuvali@cs.technion.ac.il
[3] Department of Computer Science, UCLA, Los Angeles, USA
rafail@cs.ucla.edu
[4] Department of Computer Science, RPI, Troy, USA
vzikas@cs.rpi.edu

Abstract. Traditional protocols for secure multi-party computation among n parties communicate at least a linear (in n) number of bits, even when computing very simple functions. In this work we investigate the feasibility of protocols with *sublinear* communication complexity. Concretely, we consider two clients, one of which may be corrupted, who wish to perform some "small" joint computation using n servers but without any trusted setup. We show that enforcing sublinear communication complexity drastically affects the feasibility bounds on the number of corrupted parties that can be tolerated in the setting of information-theoretic security.

We provide a complete investigation of security in the presence of semi-honest adversaries—static and adaptive, with and without erasures—and initiate the study of security in the presence of malicious adversaries. For semi-honest static adversaries, our bounds essentially match the corresponding bounds when there is no communication restriction—i.e., we can tolerate up to $t < (1/2 - \epsilon)n$ corrupted parties. For the adaptive case, however, the situation is different. We prove that without erasures even a small constant fraction of corruptions is intolerable, and—more surprisingly—when erasures are allowed, we prove that $t < (1 - \sqrt{0.5} - \epsilon)n$ corruptions can be tolerated, which we also show to be essentially optimal. The latter optimality proof hinges on a new treatment of probabilistic adversary structures that may be of independent interest. In the case of active corruptions in the sublinear communication setting, we prove that static "security with abort" is feasible when $t < (1/2 - \epsilon)n$, namely, the bound that is tight for semi-honest security. All of our negative results in fact rule out protocols with sublinear *message complexity*.

The full version of this paper can be found at the Cryptology ePrint Archive [28].

© International Association for Cryptologic Research 2017
J. Katz and H. Shacham (Eds.): CRYPTO 2017, Part I, LNCS 10401, pp. 420–446, 2017.
DOI: 10.1007/978-3-319-63688-7_14

1 Introduction

Secure multi-party computation (MPC) allows a set of parties to compute a function on their joint inputs in a secure way. Roughly speaking, security means that even when some of the parties misbehave, they can neither disrupt the output of honest parties (correctness), nor can they obtain more information than their specified inputs and outputs (privacy). Misbehaving parties are captured by assuming an adversary that corrupts some of the parties and uses them to attack the protocol. The usual types of adversary are *semi-honest* (aka "passive"), where the adversary just observes the view of corrupted parties, and *malicious* (aka "active"), where the adversary takes full control of the corrupted parties.

The seminal results from the '80s [32,52] proved that under standard cryptographic assumption, any multi-party functionality can be securely computed in the presence of a polynomially bounded semi-honest adversary corrupting arbitrarily many parties. For the malicious case, Goldreich et al. [32] proved that arbitrarily many corruptions can be tolerated if we are willing to give up on fairness, and achieve so-called *security with abort*; otherwise, an honest majority is required.

In the information-theoretic (IT) model—where there are no restrictions on the adversary's computational power—the situation is different. Ben-Or et al. [4] and independently Chaum et al. [14] proved that IT security is possible if and only if $t < n/3$ parties are actively corrupted (or $t < n/2$ are passively corrupted, respectively). The solutions in [4] are perfectly secure, i.e., there is a zero-error probability. Rabin and Ben-Or [50] proved that if a negligible error probability is allowed, and a broadcast channel is available to the parties, then any function can be IT-securely computed if and only if $t < n/2$ parties are actively corrupted. All the above bounds hold both for a *static* adversary, who chooses which parties to corrupt at the beginning of the protocol execution, and for an *adaptive* adversary, who might corrupt more parties as the protocol evolves and depending on his view of the protocol so far.

In addition to their unconditional security and good concrete efficiency, information theoretic protocols typically enjoy strong composability guarantees. Concretely, the above conditions for the IT setting allow for universally composable (UC) protocols [10]. This is known to be impossible in the *plain model*—i.e., without assuming access to a trusted setup functionality such as a common reference string (CRS) [12], even if one settles for computational security. Given the above advantages of IT protocols, it is natural to investigate alternative models that allow for IT-secure protocols without an honest majority.

It is well known that assuming a strong setup such as *oblivious transfer* (OT) [49], we can construct IT secure protocols tolerating an arbitrary number of corruptions both in the semi-honest setting [32] and in the malicious setting [43,45]. However, these solutions require trusting (a centralized party that serves as) an OT functionality.

An alternative approach is for the parties to procure help from other servers in a network they have access to, such as the Internet. This naturally leads to

the formulation of the problem in the so-called *client-server* model [16,18,19,36]. This model refines the standard MPC model by separating parties into *clients*, who wish to perform some computation and provide the inputs to and receive outputs from it, and *servers*, who help the clients perform their computation. (The same party can play both roles, as is the case in the standard model of secure computation.) The main advantage of this refinement is that it allows to decouple the number of clients from the expected "level of security," which depends on the number of servers and the security threshold, and, importantly, it allows us to address the question of how the communication complexity (CC) of the protocol increases with the number n of servers.

A direct approach to obtain security in the client/server model would be to have the clients share their input to *all* the servers (denoted by n from now on), who would perform the computation on these inputs and return to the clients their respective outputs. Using [4,14,32,50], this approach yields a protocol tolerating $t < n/2$ semi-honest corrupted servers, or, for the malicious setting, $t < n/2$ corrupted servers if broadcast is available, and $t < n/3$, otherwise. (Recall that the above bounds are required in addition to arbitrarily many corruptions of clients.)

Despite its simplicity, however, the above approach incurs a high overhead in communication when the number of clients is small in comparison to the number of servers, which is often the case in natural application scenarios. Indeed, the communication complexity of the above protocol would be polynomial in n. In this work we investigate the question of how to devise IT protocols with near-optimal resilience in the client/server model, where the communication complexity is *sublinear* in the number of servers n. As we prove, this low-communication requirement comes at a cost, inducing a different—and somewhat surprising—landscape of feasibility bounds.

Our Contributions. In this work we study the feasibility of information-theoretic MPC in the client-server model with sublinear communication complexity. We consider the case of two clients and n servers, which we refer to as the $(2, n)$-*client/server model*, and prove exact feasibility bounds on the number of corrupted servers that can be tolerated for MPC in addition to a corrupted client.[1] We provide a complete investigation of security against semi-honest adversaries—static and adaptive, with and without erasures—and also initiate the study of malicious adversaries. Our results can be summarized as follows:

– As a warmup, for the simplest possible case of static semi-honest corruptions, we confirm that the folklore protocol which has one of the clients ask a random sublinear-size server "committee" [8] to help the clients perform their computation, is secure and has sublinear message complexity against

[1] Our bounds are for the two-client case, but can be easily extended to the multi-client setting with constantly many clients, as such an extension will just incur a constant multiplicative increase in CC.

$t < (1/2 - \epsilon)n$ corrupted servers, for any given constant $0 < \epsilon < 1/2$. Further, we prove that this bound is tight. Thus, up to an arbitrarily small constant fraction, the situation is the same as in the case of MPC with unrestricted communication.

– In the case of adaptive semi-honest corruptions we distinguish between two cases, depending on whether or not the (honest) parties are allowed to erase their state. Naturally, allowing erasures makes it more difficult for the adversary to attack a protocol. However, restricting to sublinear communication complexity introduces a counterintuitive complication in providing optimally resilient protocols. Specifically, in communication-unrestricted MPC (e.g., MPC with linear or polynomial CC), the introduction of erasures does not affect the exact feasibility bound $t < n/2$ and typically makes it easier[2] to come up with a provably secure protocol against any tolerable adversary. In contrast, in the sublinear-communication realm erasures have a big effect on the feasibility bound and make the design of an optimal protocol a far more challenging task. In fact, proving upper and lower bounds for this (the erasures) setting is the most technically challenging part of this work.

  In more detail, when no erasures are assumed, we show that an adversary corrupting a constant fraction of the servers (in addition to one of the clients, say, c_1), cannot be tolerated. The reason for this is intuitive: Since there is a sublinear number of messages, there can only be a sublinear number of servers that are activated (i.e., send or receive messages) during the protocol. Thus, if the adversary has a linear corruption budget, then if he manages to find the identities of these active servers, he can adaptively corrupt all of them. Since the parties cannot erase anything (and in particular they cannot erase their communication history), the adversary corrupting c_1 can "jump" to all servers whose view depends on c_1's view, by traversing the communication graph which includes the corrupted client. Symmetrically, the adversary corrupting the other client c_2, can corrupt the remainder "protocol-relevant" parties (i.e., parties whose view depends on the joint view of the clients). Security in the presence of such an adversary contradicts classical MPC impossibility results [35], which prove that if there is a two-set partition of the party-set and the adversary might corrupt either of the sets (this is called the Q^2 condition in [35]) then this adversary cannot be tolerated for general MPC—i.e., there are functions that cannot be computed securely against such an adversary.

  Most surprising is the setting when erasures are allowed. We prove that, for any constant $\epsilon > 0$, an adversary corrupting at most $t < (1-\sqrt{0.5}-\epsilon)n$ servers can be tolerated, and moreover that this bound is essentially tight. The idea of our protocol is as follows. Instead of having the clients contact the servers for help—which would lead, as above, to the adversary corrupting too many helpers—every server probabilistically "wakes up" and volunteers to help. However, a volunteer cannot talk to both clients as with good probability the corrupted client will be the first he talks to which will result in the volunteer

[2] As opposed to requiring the use of more complex cryptographic tools such as non-committing encryption [11,21] as in the non-erasure setting.

being corrupted before erasing. Instead, each volunteer asks a random server, called *the intermediary*, to serve as his point of contact with one of the two clients. By an appropriate scheduling of message-sending and erasures, we can ensure that if the adversary jumps and corrupts a volunteer or an intermediary because he communicated with the corrupted client, then he might at most learn the message that was already sent to this client. The choice of $1 - \sqrt{0.5}$ is an optimal choice that will ensure that no adaptive adversary can corrupt more than 1/2 of the active servers set in this protocol. The intuition behind it is that if the adversary corrupts each party with probability $1 - \sqrt{0.5}$, then for any volunteer-intermediary pair, the probability that the adversary corrupts both of them before they erase (by being lucky and corrupting any on of them at random) is 1/2.

Although proving the above is far from straightforward, the most challenging part is the proof of impossibility for $t = (1 - \sqrt{0.5} + \epsilon)n$ corruptions. In a nutshell, this proof works as follows: Every adaptive adversary attacking a protocol induces a probability distribution on the set of corrupted parties; this distribution might depend on the coins of the adversary and the inputs and coins of all parties. This is because the protocol's coins and inputs define the sequence of point-to-point communication channels in the protocol, which in turn can be exploited by the adversary to expand his corruption set, by for example jumping to parties that communicate with the already corrupted set. Such a probability distribution induces a *probabilistic adversary structure* that assigns to each subset of parties the probability that this subset gets corrupted.

We provide a natural definition of what it means for such a probabilistic adversary structure to be *intolerable* and define a suitable "domination" condition which ensures that any structure that dominates an intolerable structure is also intolerable. We then use this machinery to prove that the adversary that randomly corrupts (approximately) $(1 - \sqrt{0.5})n$ servers and then corrupts everyone that talks to the corrupted parties in every protocol round induces a probabilistic structure that dominates an intolerable structure and is, therefore, also intolerable. We believe that the developed machinery might be useful for analyzing other situations in which party corruption is probabilistic.

– Finally, we initiate the study of actively secure MPC with sublinear communication. Here we look at static corruptions and provide a protocol which is IT secure with abort [32,42] against any adversary corrupting a client and $t < (1/2 - \epsilon)n$ servers for a constant $0 < \epsilon < 1/2$. This matches the semi-honest lower bound for static security, at the cost, however, of allowing the protocol to abort, a price which seems inevitable in our setting. We leave open the questions of obtaining full security or adaptive security with erasures in the case of actively secure MPC.

We finally note that both our positive and negative results are of the strongest possible form. Specifically, our designed protocols communicate a sublinear number of *bits*, whereas our impossibility proofs apply to all protocols that communicate a sublinear number of *messages* (independently of how long these messages are).

Related Work. The literature on communication complexity (CC) of MPC is vast. To put out results in perspective, we now discuss some of the most relevant literature on IT MPC with low communication complexity. For simplicity, in our discussion we shall exclude factors that depend only on the security parameter which has no dependency on n, as well as factors that are poly-logarithmic in n.

The CC of the original protocols from the '80s was polynomial (in the best case quadratic) in n, in particular, $\mathsf{poly}(n) \cdot |C|$ where $|C|$ denotes the size of the circuit C that computes the given function. A long line of work ensued that reduced this complexity down to linear in the size of the party set by shifting the dependency on different parameters [2,3,6,17,22,24–27,37–39,43,44].

In the IT setting in particular, Damgård and Nielsen [23] achieve a CC of $O(n|C|+n^2)$ messages—i.e., their CC scales in a linear fashion with the number of parties. Their protocol is perfectly secure in the presence of $t < n/2$ semi-honest corruptions. In the malicious setting, they provide a protocol tolerating $t < n/3$ corruptions with a CC of $O(n|C| + d \cdot n^2) + \mathsf{poly}(n)$ messages, where d is the multiplicative depth of the circuit C. Beerliová-Trubíniová and Hirt [3] extended this result to perfect security, achieving CC of $O(n|C| + d \cdot n^2 + n^3)$. Later on, Ben-Sasson et al. [5] achieved CC $O(n|C| + d \cdot n^2) + \mathsf{poly}(n)$ messages against $t < n/2$ active corruptions, which was brought down to $O(n|C| + n^2)$ by Genkin *et al.* [29]. Note that with the exception of the maliciously secure protocol in [23], all the above works tolerate a number of corruptions which is tight even when there is no bound on the communication complexity.

Settling for a near-optimal resilience of $t < (1/2 - \epsilon)n$, the above bounds can be improved by a factor of n, making the communication complexity grow at most polylogarithmically with the number of parties. This was first shown for client-server protocols with a constant number of clients by Damgård and Ishai [19] (see also [43]) and later in the standard MPC model by Damgård et al. [20]. The latter protocol can in fact achieve *perfect* security if $t < (1/3 - \epsilon)n$.

We point out that all the above communication bounds include polynomial (in n) *additive* terms in their CC. This means that even for circuits that are small relative to the number of parties (e.g., even when $|C| = o(n)$), they communicate a number of bits (or, worse, messages) which is polynomial in n. Instead, in this work we are interested in achieving overall (bit) communication complexity of $o(n)|C|$ without such additive (polynomial or even linear in n) terms, and are willing to settle for statistical (rather than perfect) security.

Finally, a different line of work studies the problem of reducing the *communication locality* of MPC protocols [6,7,13]. This measure corresponds to the maximum number of neighbors/parties that any party communicates with *directly*, i.e., via a bilateral channel, throughout the protocol execution. Although these

works achieve a sublinear (in n) communication locality, their model assumes each party to have an input, which requires the communication complexity to grow (at least) linearly with the number of parties. Moreover, the protocols presented in these works either assume a trusted setup or are restricted to static adversaries.

Organization of the Paper. In Sect. 2 we present the model (network, security) used in this work and establish the necessary terminology and notation. Section 3 presents our treatment of semi-honest static security, while Sect. 4 is dedicated to semi-honest adaptive corruptions, with erasures (Sect. 4.1) and without erasures (Sect. 4.2). Finally, Sect. 5 includes our feasibility result for malicious (static) adversaries.

2 Model, Definitions and Building Blocks

We consider $n+2$ parties, where two special parties, called the *clients*, wish to securely compute a function on their joint inputs with the help of the remaining n parties, called the *servers*. We denote by $\mathcal{C} = \{c_1, c_2\}$ and by $\mathcal{S} = \{s_1, \ldots, s_n\}$ the sets of clients and servers, respectively. We shall denote by $\mathcal{P}$ the set of all parties, i.e., $\mathcal{P} = \mathcal{C} \cup \mathcal{S}$. The parties are connected by a complete network of (secure) point-to-point channel as in standard unconditionally secure MPC protocols [4,14]. We call this model the $(2, n)$-*client/server model*.

The parties wish to compute a given two-party function f, described as an arithmetic circuit C_f, on inputs from the clients by invoking a synchronous protocol Π. (Wlog, we assume that f is a public-output function $f(x_1, x_2) = y$, where x_i is c_i's input; using standard techniques, this can be extended to multi-input and private-output functions—cf. [46].) Such a protocol proceeds in synchronous rounds where in each round any party might send messages to other parties and the guarantee is that any message sent in some round is delivered by the beginning of the following round. Security of the protocol is defined as security against an adversary that gets to corrupt parties and uses them to attack the protocol. We will consider both a *semi-honest* (aka *passive*) and a *malicious* (aka *active*) adversary. A semi-honest adversary gets to observe the view of parties it corrupts—and attempts to extract information from it—but allows parties to correctly execute their protocol. In contrast, a malicious adversary takes full control of corrupted parties. Furthermore, we consider both *static* and *adaptive* corruptions. A static adversary chooses the set of corrupted parties at the beginning of the protocol execution, whereas and adaptive adversary chooses this set dynamically by corrupting (additional) parties as the protocol evolves (and depending on his view of the protocol). A *threshold* (t_c, t_s)-*adversary* in the client/server model is an adversary that corrupts in total up to t_c clients and additionally up to t_s servers.

The adversary is *rushing* [9,40], i.e., in each round he first receives the messages that are sent to corrupted parties, and then has the corrupted parties send their messages for that round. For adaptive security *with erasures* we adopt

the natural model in which each of the operations "send-message," "receive-message," and "erase-messages from state" is atomic and the adversary is able to corrupt after any such atomic operation. This, in particular, means that when a party sends a message to a corrupted party, then the adversary can corrupt the sender before he erases this message. In more detail, every round is partitioned into "mini-rounds," where in each mini-round the party can send a message, or receive a message, or erase a message from its state—exclusively. This is not only a natural erasure model, but ensures that one does not design protocols whose security relies on the assumption that honest parties can send and erase a message simultaneously, as an atomic operation (see [40] for a related discussion about atomicity of sending messages).

The *communication complexity* (CC) of a protocol is the number of bits sent by honest parties during a protocol execution.[3] Throughout this work we will consider sublinear-communication protocols, i.e., protocols in which the honest (and semi-honest) parties send at most $o(n)|C_f|$ number of messages, were the message size is independent of n. Furthermore, we will only consider information-theoretic security (see below).

Simulation-Based Security. We will use the standard simulation-based definition of security from [9]. At a high level, a protocol for a given function is rendered secure against a given class of adversaries if for any adversary in this class, there exists a simulator that can emulate, in an ideal evaluation experiment, the adversary's attack to the protocol. In more detail, the simulator participates in an ideal evaluation experiment of the given function, where the parties have access to a trusted third party—often referred to as the *ideal functionality*—that receives their inputs, performs the computation and returns their outputs. The simulator takes over ("corrupts") the same set of parties as the adversary does (statically or adaptively), and has the same control as the (semi-honest or malicious) adversary has on the corrupted parties. His goal is to simulate the view of the adversary and choose inputs for corrupted parties so that for any initial input distribution, the joint distribution of the honest parties' outputs and adversarial view in the protocol execution is indistinguishable from the joint distribution of honest outputs and the simulated view in an ideal evaluation of the function. Refer to [9] for a detailed specification of the simulation-based security definition.

In this work we consider information-theoretic security and therefore we will require statistical indistinguishability. Using the standard definitions of *negligible functions* [30], we say that a pair of distribution ensembles $\mathcal{X}$ and $\mathcal{Y}$ indexed by $n \in \mathbb{N}$ are *(statistically) indistinguishable* if for all (not necessarily efficient) distinguishers D the following function with domain S:

[3] Note that in the semi-honest setting this number equals the total number of bits received during the protocol. However, in the malicious setting, corrupted parties might attempt to send more bits to honest parties than what the protocol specifies, thereby flooding the network and increasing the total number of bits received. As we shall see, our malicious protocol defends even against such an attack by having the parties abort if they receive too many bits/messages.

$$\Delta_{\mathcal{X},\mathcal{Y}}(n) := |\Pr[D(\mathcal{X}_n) = 1] - \Pr[D(\mathcal{Y}_n) = 1]|$$

is negligible in s. In this case we write $\mathcal{X} \approx \mathcal{Y}$ to denote this relation. We will further use $\mathcal{X} \equiv \mathcal{Y}$ to denote the fact that $\mathcal{X}$ and $\mathcal{Y}$ are identically distributed.

The view of the adversary in an execution of a protocol consists of the inputs and randomness of all corrupted parties and all the messages sent and received during the protocol execution. We will use $\textsc{View}_{\mathcal{A},\Pi}$ to denote the random variable (ensemble) corresponding to the view of the adversary when the parties run protocol Π. The view $\textsc{View}_{\sigma,f}$ of the simulator σ in an ideal evaluation of f is defined analogously.

For a probability distribution Pr over a sample space $\mathcal{T}$ and for any $T \in \mathcal{T}$ we will denote by $\Pr(T)$ the probability of T. We will further denote by $T \leftarrow \Pr$ the action of sampling the set T from the distribution Pr. In slight abuse of notation, for an event E we will denote by $\Pr(E)$ the probability that E occurs. Finally, for random variables $\mathcal{X}$ and $\mathcal{Y}$ we will denote by $\Pr_{\mathcal{X}}(x)$ the probability that $\mathcal{X} = x$ and by $\Pr_{\mathcal{X}|\mathcal{Y}}(x|y)$ the probability that $\mathcal{X} = x$ conditioned on $\mathcal{Y} = y$.

Oblivious Transfer and OT Combiners. Oblivious Transfer (OT) [49] is a two-party functionality between a *sender* and a *receiver*. In its most common variant called 1-out-of-2-OT,[4] the sender has two inputs $x_0, x_1 \in \{0,1\}$ and the receiver has one bit input $b \in \{0,1\}$, called the *selection* bit. The functionality allows the sender to transmit the input x_b to the receiver so that (1) the sender does not learn which bit was transmitted (i.e., learns nothing), and (2) the receiver does not learn anything about the input $x_{\bar{b}}$.

As proved by Kilian and Goldreich et al. [32,45], the OT primitive is *complete* for secure xtwo-party computation (2PC), even against malicious adversaries. Specifically, Kilian's result shows that given the ability to call an ideal oracle/functionality f_{OT} that computes OT, two parties can securely compute an arbitrary function of their inputs with unconditional security. The efficiency of these protocols was later improved by Ishai *et al.* [43].

Beaver [1] showed how OT can be pre-computed, i.e., how parties can, in an offline phase, compute correlated randomness that allows, during the online phase, to implement OT by simply the sender sending to the receiver two messages of the same length as the messages he wishes to input to the OT hybrid (and the receiver sending no message). Thus, a trusted party which is equivalent (in terms of functionality) to OT, is one that internally pre-computes the above correlated randomness and hands to the sender and the receiver their "parts" of it. We will refer to such a correlated randomness setup where the sender receives R_s and the receiver R_r as an (R_s, R_r) *OT pair*. The size of each component in such an OT pair is the same as (or linear in) the size of the messages (inputs) that the parties would hand to the OT functionality.

A fair amount of work has been devoted to so-called *OT combiners*, namely, protocols that can access several, say, m OT protocols, out of which ℓ might be insecure, and combine them into a secure OT protocol (e.g., [33,34,47]). OT

[4] In this work we will use OT to refer to 1-out-of-2 OT.

combiners with linear rate (i.e., where the total communication of the combiner is linear in the total communication of the OT protocol) exist both for semi-honest and for malicious security as long as $\ell < m/2$. Such an OT combiner can be applied to the pre-computed OT protocol to transform m precomputed OT strings out of which ℓ are sampled from the appropriate distribution by a trusted party, into one securely pre-computed OT string, which can then be used to implement a secure instance of OT.

3 Sublinear Communication with Static Corruptions

As a warm up, we start our treatment of secure computation in the $(2, n)$-client/server model with the case of a static adversary, where, as we show, requiring sublinear communication complexity comes almost at no cost in terms of how many corrupted parties can be tolerated. We consider the case of a semi-honest adversary and confirm that using a "folklore" protocol any $(1, t)$-adversary with $t < (\frac{1}{2} - \epsilon)n$ corruptions can be tolerated, for an arbitrary constant $0 < \epsilon < \frac{1}{2}$. We further prove that this bound is tight (up to an arbitrary small constant fraction of corruptions); i.e., if for some $\epsilon > 0, t = (\frac{1}{2} + \epsilon)n$, then a semi-honest $(1, t)$-adversary cannot be tolerated.[5]

Specifically, in the static semi-honest case the following folklore protocol based on the approach of selecting a random committee [8] is secure and has sublinear message complexity. This protocol has any of the two clients, say, c_1, choose (with high probability) a random committee/subset of the servers of at most polylogarithmic size and inform the other client about his choice. These servers are given as input secret sharings of the clients' inputs, and are requested to run a standard MPC protocol that is secure in the presence of an honest majority, for example, the semi-honest MPC protocol by Ben-Or et al. [4], hereafter referred to as the "BGW" protocol. The random choice of the servers that execute the BGW protocol will ensure that, except with negligible (in n) probability, a majority of them will be honest. Furthermore, because the BGW protocol's complexity is polynomial in the party size, which in this case is polylogarithmic, the total communication complexity in this case is polylogarithmic. We denote the above protocol as Π_{stat} and state its security in Theorem 1. The proof is simple and follows the above idea. We refer to the full version [28] for details.

Theorem 1. *Protocol Π_{stat} unconditionally securely computes any given 2-party function f in the $(2, n)$-client/server model in the presence of a passive and static $(1, t)$-adversary with $t < (1/2 - \epsilon)n$, for any given constant $0 < \epsilon < 1/2$. Moreover, Π_{stat} communicates $O(\log^{\delta'}(n)|C_f|)$ messages, for a constant $\delta' > 1$.*

Next, we prove that Theorem 1 is tight. The proof idea is as follows: If the adversary can corrupt a majority of the servers, i.e., $t \geq n/2$, then no matter which subset of the servers is actually activated (i.e., sends or receives a

[5] Wlog we can assume that the semi-honest adversary just outputs his entire view [9]; hence semi-honest adversaries only differ in the set of parties they corrupt.

message) in the protocol[6], an adversary that randomly chooses the parties to corrupt has a good chance of corrupting any half of the active server set. Thus, existence of a protocol for computing, e.g., the OR function while tolerating such an adversary would contradict the impossibility result by Hirt and Maurer [35] which implies that an adversary who can corrupt a set and its complement—or supersets thereof—is intolerable for the OR function. The actual theorem statement is tighter, and excludes even adversaries that corrupt $t \geq n/2 - \delta$, for some constant $\delta \geq 0$. The proof uses the above idea with the additional observation that due to the small (sublinear) size of the set $\bar{\mathcal{S}}$ of *active* servers, i.e., servers that send or receive a message in the protocol, a random set of $\delta = O(1)$ servers has noticeable chance to include no such active server. We refer to the full version of this work [28] for a formal proof.

Theorem 2. *Assuming a static adversary, there exists no information theoretically secure protocol for computing the boolean OR of the (two) clients' inputs with message complexity $m = o(n)$ tolerating a $(1, t)$-adversary with $t \geq n/2 - \delta$, for some $\delta = O(1)$.*

4 Sublinear Communication with Adaptive Corruptions

In this section we consider an adaptive semi-honest adversary and prove corresponding tight bounds for security with erasures—the protocol can instruct parties to erase their state so as to protect information from an adaptive adversary who has not yet corrupted the party—and without erasures—everything that the parties see stays in their state.

4.1 Security with Erasures

We start with the setting where erasures of the parties' states are allowed, which prominently demonstrates that sublinear communication comes at an unexpected cost in the number of tolerable corruptions. Specifically, in this section we show that for any constant $0 < \epsilon < 1 - \sqrt{0.5}$, there exists a protocol that computes any given two-party function f in the presence of a $(1, t)$-adversary if $t < (1 - \sqrt{0.5} - \epsilon)n$ (Theorem 3). Most surprisingly, we prove that this bound is tight up to any arbitrary small constant fraction of corruptions (Theorem 4). The technique used in proving the lower bound introduces a novel treatment of (and a toolboox for) probabilistic adversary structures that we believe can be of independent interest.

We start with the protocol construction. First, observe that the idea behind protocol Π_{stat} cannot work here as an adaptive adversary can corrupt client c_1, wait for him to choose the servers in $\bar{\mathcal{S}}$, and then corrupt all of them adaptively since he has a linear corruption budget. (Note that erasures cannot help here as the adversary sees the list of all receivers by observing the corrupted

[6] Note that not all servers can be activated as the number of active servers is naturally bounded by the (sublinear) communication complexity.

sender's state.) This attack would render any protocol non-private. Instead, we will present a protocol which allows clients c_1 and c_2 to pre-compute sufficiently many 1-out-of-2 OT functionalities $f_{\mathrm{OT}}((m_0, m_1), b) = (\perp, m_b)$ in the $(2, n)$-client/server model with sublinear communication complexity. The completeness of OT ensures that this allows c_1 and c_2 to compute any given function.

A first attempt towards the above goal is as follows. Every server independently decides with probability $p = \frac{\log^\delta n}{n}$ (based on his own local randomness) to "volunteer" in helping the clients by acting as an OT dealer (i.e., acting as a trusted party that prepares and sends to the clients an OT pair). The choice of p can be such that with overwhelming probability not too many honest servers volunteer (at most sublinear in n) and the majority of the volunteers are honest. Thus, the majority of the distributed OT pairs will be honest, which implies that the parties can use an OT-combiner that is secure for a majority of good OTs (e.g., [34]) on the received OT pairs to derive a secure implementation of OT.

Unfortunately, the above idea does not quite work. To see why, consider an adversary who randomly corrupts one of the clients and as soon as any honest volunteer sends a messages to the corrupted client, the adversary corrupts him as well and reads his state. (Recall that send and erase are atomic operations.) It is not hard then to verify that even if the volunteer erases part of its state between contacting each of the two clients, with probability (at least) 1/2 such an adversary learns the entire internal state of the volunteer before he gets a chance to erase it.

So instead of the above idea, our approach is as follows. Every server, as above, decides with probability $p = \frac{\log^\delta n}{n}$ to volunteer in helping the clients by acting as an OT dealer and computes the OT pair, *but does not send it*. Instead, it first chooses another server, which we refer to as his *intermediary*, uniformly at random, and forwards him one of the components in the OT pairs (say, the one intended for the receiver); then, it erases the sent component and the identity of the intermediary along with the coins used to sample it (so that now his state only includes the sender's component of the OT pair); finally, both the volunteer and his intermediary forward their values to their intended recipient.

It is straightforward to verify that with the above strategy the adversary does not gain anything by corrupting a helping server—whether a volunteer or his associated intermediary—when he talks to the corrupted client. Indeed, at the point when such a helper contacts the client, the part of the OT pair that is not intended for that client and the identity of the other associated helper have both been erased. But now we have introduced an extra point of possible corruption: The adversary can learn any given OT pair by corrupting either the corresponding volunteer or his intermediary before the round where the clients are contacted. However, as we will show, when $t < (1-\sqrt{0.5}-\epsilon)n$, the probability that the adversary corrupts more than half of such pairs is negligible.

The complete specification of the above sketched protocol, denoted $\Pi^{\mathrm{OT}}_{\mathrm{adap}}$, and the corresponding security statement are shown below.

Protocol $\Pi^{\mathrm{OT}}_{\mathrm{adap}}(\mathcal{C} = \{c_1, c_2\}, \mathcal{S} = \{s_1, \ldots, s_n\})$

1. Every server $s_i \in \mathcal{S}$ locally decides to become active with probability $p = \frac{\log^\delta n}{n}$ for a publicly known constant $\delta > 1$. Let $\bar{\mathcal{S}}_1$ denote the set of parties that become active in this round. Every $s_i \in \bar{\mathcal{S}}_1$ prepares an OT pair $((m_i, r_i), \mathtt{otid}_i)$, where $\mathtt{otid}_i \in \{0,1\}^{\log^\delta n}$ is a uniformly chosen identifier.
2. Every $s_i \in \bar{\mathcal{S}}_1$ choses an intermediary $s_{ij} \in \mathcal{S}$ uniformly at random and sends $(r_i, \mathtt{otid}_i)$ to s_{ij}. Denote by $\bar{\mathcal{S}}_2 = \{s_{ij} | s_i \in \bar{\mathcal{S}}\}$ the set of all relayers.
3. Every $s_i \in \bar{\mathcal{S}}_1$ erases r_i, the identity of s_{ij}, and the randomness used to select s_{ij}; and every $s_{ij} \in \bar{\mathcal{S}}_2$ erases the identity of s_i.
4. Every $s_i \in \bar{\mathcal{S}}_1$ sends $(m_i, \mathtt{otid}_i)$ to c_1 and every $s_{ij} \in \bar{\mathcal{S}}_2$ sends $(r_i, \mathtt{otid}_i)$ to c_2.
5. Every $s_i \in \bar{\mathcal{S}}_1$ and every $s_{ij} \in \bar{\mathcal{S}}_2$ erase their entire internal state.
6. The clients c_1 and c_2 use the OT pairs with matching $\mathtt{otid}$'s within a (semi-honest) $(n/2, n)$ OT-combiner [34] to obtain a secure OT protocol.

Theorem 3. *Protocol Π^{OT}_{adap} unconditionally securely computes the function $f_{OT}((m_0, m_1), b) = (\perp, m_b)$ in the $(2, n)$-client/server model in the presence of a passive and adaptive $(1, t)$-adversary with $t < (1 - \sqrt{0.5} - \epsilon)n$, for any given constant $0 < \epsilon < 1 - \sqrt{0.5}$ and assuming erasures. Moreover, Π^{OT}_{adap} communicates $O(\log^\delta(n))$ messages, with $\delta > 1$, except with negligible probability.*

Proof. Every server $s \in \mathcal{S}$ is included in the set of servers that become active in the first round, i.e., $\bar{\mathcal{S}}_1$, with probability $p = \frac{\log^\delta n}{n}$ independent of the other servers. Thus by application of the Chernoff bound we get that for every $0 < \gamma < 1/2$:

$$\Pr[|\bar{\mathcal{S}}_1| > (1+\gamma)\log^\delta n] < e^{-\frac{\gamma \log^\delta n}{3}} \tag{1}$$

which is negligible. Moreover, each $s_i \in \bar{\mathcal{S}}_1$ chooses one additional relay-party s_{ij} which means that for any constant $1/2 < \gamma' < 1$:

$$|\bar{\mathcal{S}}| = |\bar{\mathcal{S}}_1 \cup \bar{\mathcal{S}}_2| \leq (2 + \gamma')\log^\delta n$$

with overwhelming probability. (As in the proof of Theorem 2, $\bar{\mathcal{S}}$ denotes the set of active servers at the end of the protocol.) Since each such party communicates at most two messages, the total message complexity is $O(\log^\delta n)$ plus the messages exchanged in the OT combiner which are polynomial in the number of OT pairs. Thus, with overwhelming probability, the total number of messages is $O(\log^{\delta'}(n))$ for some constant $\delta' > \delta$.

To prove security, it suffices to ensure that for the uncorrupted client, the adversary does not learn at least half of the received OT setups. Assume wlog that c_2 is corrupted. (The case of a corrupted c_1 is handled symmetrically, because, wlog, we can assume that an adversary corrupting some party in $\bar{\mathcal{S}}_1$ also corrupts all parties in $\bar{\mathcal{S}}_2$ which this party sends messages to after its corruption.) We show that the probability that the adversary learns more than half of the m_i's is negligible.

First, we can assume, wlog, that the adversary does not corrupt any servers after Step 5, i.e., after the states of the servers have been erased. Indeed, for any such adversary $\mathcal{A}$ there exists an adversary $\mathcal{A}'$ who outputs a view with the same distribution as $\mathcal{A}$ but does not corrupt any of the parties that $\mathcal{A}$ corrupts after Step 5; in particular $\mathcal{A}'$ uses $\mathcal{A}$ as a blackbox and follows $\mathcal{A}$'s instructions, and until Step 5 corrupts every server that $\mathcal{A}$ requests to corrupt, but after that step, any request from $\mathcal{A}$ to corrupt a new server s is replied by $\mathcal{A}'$ simulating s without corrupting him. (This simulation is trivially perfect since at Step 5, s will have erased its local state so $\mathcal{A}'$ needs just to simulate the unused randomness.)

Second, we observe that, since the adversary does not corrupt c_1, the only way to learn some m_i is by corrupting the party in $\bar{\mathcal{S}}_1$ that sent it to c_1. Hence to prove that the adversary learns less than 1/2 of the m_i's it suffices to prove that the adversary corrupts less than 1/2 of $\bar{\mathcal{S}}_1$.

Next, we observe that the adversary does not gain any advantage in corrupting parties in $\bar{\mathcal{S}}_1$ by corrupting client c_2, since (1) parties in $\bar{\mathcal{S}}_1$ do not communicate with c_2, and (2) by the time an honest party $s_{ij} \in \bar{\mathcal{S}}_2$ communicates with c_2 he has already erased the identity of s_i. (Thus, corrupting s_{ij} after he communicates with c_2 yields no advantage in finding s_i.) Stated differently, if there is an adversary who corrupts more than 1/2 servers in $\bar{\mathcal{S}}_1$, then there exists an adversary that does the same without even corrupting c_2. Thus, to complete the proof it suffices to show that any adversary who does not corrupt c_2, corrupts less than 1/2 of the servers in $|\bar{\mathcal{S}}_1|$. This is stated in Lemma 2, which is proved using the following strategy: First, we isolate a "bad" subset $\bar{\mathcal{S}}_1'$ of $\bar{\mathcal{S}}_1$ which we call *over-connected parties,* for which we cannot give helpful guarantees on the number of corruptions. Nonetheless, we prove in Lemma 1 that this "bad" set is "sufficiently small" compared to $\bar{\mathcal{S}}_1$. By this we mean that we can bound the fraction of corrupted parties in $\bar{\mathcal{S}}_1$ sufficiently far from 1/2 so that even if give this bad set $\bar{\mathcal{S}}_1'$ to the adversary to corrupt for free, his chances of corrupting a majority in $\bar{\mathcal{S}}_1$ are still negligible. The formal arguments follow.

Let $E = \{(s, s') \mid s \in \bar{\mathcal{S}}_1 \vee s' \in \bar{\mathcal{S}}_2\}$ and let G denote the graph with vertex-set $\mathcal{S}$ and edge-set E. We say that server $s_i \in \bar{\mathcal{S}}_1$ is an *over-connected* server if the set $\{s_i, s_{ij}\}$ has neighbors in G. Intuitively, the set of over-connected servers is chosen so that if we remove these servers from G we get a perfect matching between $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$.

Next, we show that even if we give up all over-connected servers in $\bar{\mathcal{S}}_1$ (i.e., allow the adversary to corrupt all of them for free) we still have a majority of uncorrupted servers in $\bar{\mathcal{S}}_1$. For this purpose, we first prove in Lemma 1 that the fraction of $\bar{\mathcal{S}}_1$ servers that are over-connected is an arbitrary small constant.

Lemma 1. *Let $\bar{\mathcal{S}}_1' \subseteq \bar{\mathcal{S}}_1$ denote the set of over-connected servers as defined above. For any constant $1 > \epsilon' > 0$ and for large enough n, $|\bar{\mathcal{S}}_1'| < \epsilon'|\bar{\mathcal{S}}_1|$ except with negligible probability.*

Proof. To prove the claim we make use of the generalized Chernoff bound [48]. For each $s_i \in \bar{\mathcal{S}}_1$ let $X_i \in \{0, 1\}$ denote the indicator random variable that is 1 if $s_i \in \bar{\mathcal{S}}_1'$ and 0 otherwise. As above for each $s_i \in \bar{\mathcal{S}}_1$ we denote by s_{ij} the party that s_i chooses as its intermediary in the protocol.

$$\begin{aligned}\Pr[X_i = 1] &= \Pr[(s_{ij} \in \bar{\mathcal{S}}_1) \cup (\exists s_k \in \bar{\mathcal{S}}_1 : s_{kj} \in \{s_i, s_{ij}\})] \\ &\leq \Pr[s_{ij} \in \bar{\mathcal{S}}_1] + \Pr[\exists s_k \in \bar{\mathcal{S}}_1 : s_{kj} = s_i] \\ &\qquad + \Pr[\exists s_k \in \bar{\mathcal{S}}_1 : s_{kj} = s_{ij}] \\ &\leq 3\frac{|\bar{\mathcal{S}}_1|}{n},\end{aligned}$$

where both inequalities follow by a direct union bound since s_{ij} is chosen uniformly at random, and for each of the servers s_i and s_{ij} there are at most $|\bar{\mathcal{S}}_1|$ servers that might choose them as an intermediary. But from Eq. 1, $|\bar{\mathcal{S}}_1| < (1+\gamma)\log^{\delta} n$ except with negligible probability. Thus, for large enough n, $\Pr[X_i = 1] < \epsilon'$.

Next, we observe that for any subset Q of indices of parties in $\bar{\mathcal{S}}_1$ and for any $i \in Q$ it holds that $\Pr[X_i = 1 \mid \bigwedge_{j \in Q \setminus \{i\}} X_j = 1] \leq \Pr[X_i = 1]$. This is the case because the number of edges (s_k, s_{kj}) is equal to the size of $\bar{\mathcal{S}}_1$ and any connected component in G with ℓ nodes must include at least ℓ such edges. Hence, for any such Q, $\Pr[\wedge_{i \in Q} X_i = 1] \leq \prod_{i \in Q} \Pr[X_i = 1] \leq \epsilon_1^{|Q|}$. Therefore, by an application of the generalized Chernoff bound [48], for $\delta = \epsilon_1 < \epsilon'$ and $\gamma = \epsilon'$, we obtain

$$\Pr[\sum_{i=1}^{n} X_i \geq \epsilon' n] \leq e^{-n2(\epsilon' - \epsilon_1)^2},$$

which is negligible. □

Now, let $\mathcal{A}$ be an adaptive $(1,t)$-adversary and let C be the total set of servers corrupted by $\mathcal{A}$ (at the end of Step 5). We want to prove that $|C \cap \bar{\mathcal{S}}_1| < \frac{1}{2}|\bar{\mathcal{S}}_1|$ except with negligible probability. Towards this objective, we consider the adversary $\mathcal{A}'$ who is given access to the identities of all servers in $\bar{\mathcal{S}}'_1$, corrupts all these parties and, additionally, corrupts the first $t - |\bar{\mathcal{S}}'_1|$ parties that adversary $\mathcal{A}$ corrupts. Let C' denote the set of parties that $\mathcal{A}'$ corrupts. It is easy to verify that if $|C \cap \bar{\mathcal{S}}_1| \geq \frac{1}{2}|\bar{\mathcal{S}}_1|$ then $|C' \cap \bar{\mathcal{S}}_1| \geq \frac{1}{2}|\bar{\mathcal{S}}_1|$. Indeed, $\mathcal{A}'$ corrupts all but the last $|\bar{\mathcal{S}}'_1|$ of the parties that $\mathcal{A}$ corrupts; if all these last parties end up in $\bar{\mathcal{S}}_1$ then we will have $|C' \cap \bar{\mathcal{S}}_1| = |C \cap \bar{\mathcal{S}}_1|$, otherwise, at least one of them will not be in $C \cap \bar{\mathcal{S}}_1$ in which case we will have $|C' \cap \bar{\mathcal{S}}_1| > |C \cap \bar{\mathcal{S}}_1|$. Hence, to prove that $|C \cap \bar{\mathcal{S}}_1| < \frac{1}{2}|\bar{\mathcal{S}}_1|$ it suffices to prove that $|C' \cap \bar{\mathcal{S}}_1| < \frac{1}{2}|\bar{\mathcal{S}}_1|$.

Lemma 2. *The set C' of servers corrupted by $\mathcal{A}'$ as above has size $|C' \cap \bar{\mathcal{S}}_1| < \frac{1}{2}|\bar{\mathcal{S}}_1|$, except with negligible probability.*

Proof. Consider the gaph G' which results by deleting from G the vertices/servers in $\bar{\mathcal{S}}'_1$. By construction, G' is a perfect pairing between parties in $\bar{\mathcal{S}}_1 \setminus \bar{\mathcal{S}}'_1$ and parties in $\bar{\mathcal{S}}_2 \setminus \bar{\mathcal{S}}'_1$. For each $s_i \in \bar{\mathcal{S}}_1 \setminus \bar{\mathcal{S}}'_1$, let X_i denote the Boolean random variable with $X_i = 1$ if $\{s_i, s_{ij}\} \cap (C' \setminus \bar{\mathcal{S}}'_1) \neq \emptyset$ and $X_i = 0$ otherwise. When $X_i = 1$, we say that the adversary has corrupted the edge $e_i = (s_i, s_{ij})$. Clearly, the number of corrupted edges is an upper bound on the corresponding number of corrupted servers in $\bar{\mathcal{S}}_1 \setminus \bar{\mathcal{S}}'_1$. Thus, we will show that the number of corrupted edges is bounded away from 1/2.

By construction of G' the X_i's are independent, identically distributed random variables. Every edge in G' is equally likely, thus the adversary gets no information on the rest of the graph by corrupting some edge. Therefore we can assume wlog that $\mathcal{A}'$ chooses the servers in $C' \setminus \bar{\mathcal{S}}'_1$ at the beginning of the protocol execution. In this case we get the following for $C'_1 = C' \setminus \bar{\mathcal{S}}'_1$:

$$\begin{aligned}\Pr[X_i = 1] &= \Pr[s_i \in C'_1] + \Pr[s_{ij} \in C'_1] - \Pr[\{s_i, s_{ij}\} \subseteq C'_1] \\ &= 2\frac{|C| - |\bar{\mathcal{S}}'_1|}{n - |\bar{\mathcal{S}}'_1|} - \left(\frac{|C| - |\bar{\mathcal{S}}'_1|}{n - |\bar{\mathcal{S}}'_1|}\right)^2 \\ &\leq \frac{2(1 - \sqrt{0.5} - \epsilon)n}{n - |\bar{\mathcal{S}}'_1|} - \left(\frac{(1 - \sqrt{0.5} - \epsilon)n - |\bar{\mathcal{S}}'_1|}{n - |\bar{\mathcal{S}}'_1|}\right)^2.\end{aligned}$$

To make the notation more compact, let $\lambda = 1 - \sqrt{0.5} - \epsilon$. Because, from Lemma 1, $|\bar{\mathcal{S}}'_1| \leq \epsilon' n$ (and thus $n - |\bar{\mathcal{S}}'_1| > (1 - \epsilon')n$) except with negligible probability, we have that for large enough n and some negligible function μ:

$$\Pr[X_i = 1] \leq \frac{2\lambda n}{(1 - \epsilon')n} - \left(\frac{\lambda n - |\bar{\mathcal{S}}'_1|}{n - |\bar{\mathcal{S}}'_1|}\right)^2 + \mu. \tag{2}$$

Moreover,

$$\begin{aligned}\left(\frac{\lambda n - |\bar{\mathcal{S}}'_1|}{n - |\bar{\mathcal{S}}'_1|}\right)^2 &\geq \left(\frac{\lambda n - |\bar{\mathcal{S}}'_1|}{n}\right)^2 = \left(\lambda - \frac{|\bar{\mathcal{S}}'_1|}{n}\right)^2 \\ &\geq \lambda^2 - \frac{2\lambda|\bar{\mathcal{S}}'_1|}{n}.\end{aligned} \tag{3}$$

But because, from Eq. 1, $|\bar{\mathcal{S}}_1| = O(\log^\delta n)$ with overwhelming probability, we have that for every constant $0 < \epsilon_1 < 1$ and every negligible function μ', and for all sufficiently large n, $\frac{2\lambda|\bar{\mathcal{S}}'_1|}{n} + \mu' < \epsilon_1$ holds. Thus, combining Eqs. 2 and 3 we get that for all such ϵ_1 and for sufficiently large n:

$$\begin{aligned}\Pr[X_i = 1] &\leq \frac{2}{(1 - \epsilon')}\lambda - \lambda^2 + \epsilon_1 \\ &= \frac{2}{(1 - \epsilon')}(1 - \sqrt{0.5} - \epsilon) - 1.5 - \epsilon^2 + 2\epsilon + 2(1 - \epsilon)\sqrt{0.5} + \epsilon_1 \\ &\leq \frac{2}{(1 - \epsilon')} - \frac{2\epsilon}{(1 - \epsilon')} - 1.5 - \epsilon^2 + 2\epsilon + \epsilon_1 \\ &\leq \frac{2}{(1 - \epsilon')} - 1.5 - \epsilon^2 + \epsilon_1.\end{aligned}$$

For $\epsilon' \leq 1 - \frac{2}{2+\epsilon^2/4}$ and $\epsilon_1 = \epsilon^2/4$, the last equation gives

$$\Pr[X_i = 1] \leq \frac{1}{2} - \frac{\epsilon^2}{2}.$$

Furthermore, because the X_i's are independent, the assumptions in [48] are satisfied for $\delta = \frac{1}{2} - \frac{\epsilon^2}{2}$, hence,

$$\Pr[\sum_{s_i \in \bar{\mathcal{S}}_1 \setminus \bar{\mathcal{S}}'_1} X_i \geq (1/2 - \epsilon^2/3)|\bar{\mathcal{S}}_1 \setminus \bar{\mathcal{S}}'_1|] \leq e^{-n(\epsilon^2/6)},$$

which is negligible. Note that, by Lemma 1, for large enough n, with overwhelming probability $|\bar{\mathcal{S}}'_1| < \frac{2\epsilon^2}{3+2\epsilon^2}|\bar{\mathcal{S}}_1|$. Thus, with overwhelming probability the total number of corrupted servers in $\bar{\mathcal{S}}_1$ is less than $\frac{1}{2}|\bar{\mathcal{S}}_1|$. □

The above lemma ensures that the adversary cannot corrupt a majority of the OT pairs. Furthermore, with overwhelming probability, all the `otid`'s chosen by the parties in $\bar{\mathcal{S}}$ are distinct. Thus, the security of the protocol follows from the security of the OT combiner. This concludes the proof of Theorem 3. □

Next, we turn to the proof of the lower bound. We prove that there exists an adaptive $(1, t)$-adversary that cannot be tolerated when $t = (1 - \sqrt{0.5} + \epsilon)n$ for any (arbitrarily small) constant $\epsilon > 0$. To this end, we start with the observation that every adaptive adversary attacking a protocol induces a probability distribution on the set of corrupted parties, which might depend on the coins of the adversary and on the inputs and coins of all parties. Such a probability distribution induces a *probabilistic adversary structures* that assigns to each subset of parties the probability that this subset gets corrupted. Hence, it suffices to prove that this probabilistic adversary structure is what we call *intolerable*, which, roughly, means that there are functions that cannot be computed when the corrupted sets are chosen from this structure. Before sketching our proof strategy, it is useful to give some intuition about the main challenge one encounters when attempting to prove such a statement. This is best demonstrated by the following counterexample.

A Counterexample. It is tempting to conjecture that for every probabilistic adversary $\mathcal{A}$ who corrupts each party i with probability $p_i > 1/2$, there is no (general purpose) information-theoretic MPC protocol which achieves security against $\mathcal{A}$. While this is true if the corruption probabilities are independent, we show that this is far from being true in general.

Let f_k denote the boolean function $f_k : \{0,1\}^{3^k} \to \{0,1\}$ computed by a depth-k complete tree of 3-input majority gates. It follows from [15,36] that there is a perfectly secure information-theoretic MPC protocol that tolerates every set of corrupted parties T whose characteristic vector χ_T satisfies $f(\chi_T) = 0$. We show the following.

Proposition 1. *There exists a sequence of distributions X_k, where X_k is distributed over $\{0,1\}^{3^k}$, such that for every positive integer k we have (1) $f_k(X_k)$ is identically 0, and (2) each entry of X_k takes the value 1 with probability $1 - (2/3)^k$.*

Proof. Define the sequence X_k inductively as follows. X_1 is a uniformly random over $\{100, 010, 001\}$. The bit-string X_k is obtained as follows. Associate the entries of X_k with the leaves of a complete ternary tree of depth k. Randomly pick X_k by assigning 1 to all leaves of one of the three sub-trees of the root (the identity of which is chosen at random), and assigning values to each of the two other sub-trees according to X_{k-1}. Both properties can be easily proved by induction on k. □

Letting $\mathcal{A}_k$ denote the probabilistic adversary corresponding to X_k, we get a strong version of the desired counterexample, thus contradicting the aforementioned conjecture for $k \geq 2$.

The above counterexample demonstrates that even seemingly straightforward arguments when considering probabilistic adversary structures can be false, because of correlation in the corruption events. Next, we present the high-level structure of our lower bound proof.

We consider an adversary $\mathcal{A}$ who works as follows: At the beginning of the protocol, $\mathcal{A}$ corrupts each of the n servers independently with probability $1-\sqrt{0.5}$ and corrupts one of the two clients, say, c_1, at random; denote the set of initially corrupted servers by C_0 and initialize $C := C_0$. Subsequently, in every round, if any server sends or/receives a message to/from one of the servers in C, then the adversary corrupts him as well and adds him to C. Observe that $\mathcal{A}$ does *not* corrupt servers when they send or receive messages to the clients. (Such an adversary would in fact be stronger but we will show that even the above weaker adversary cannot be tolerated.) We also note that the above adversary might exceed his corruption budget $t = (1 - \sqrt{0.5} - \epsilon)n$. However, an application of the Chernoff bound shows that the probability that this happens in negligible in n so we can simply have the adversary abort in the unlikely case of such an overflow.

We next observe that because $\mathcal{A}$ corrupts servers independently at the beginning of the protocol, we can consider an equivalent random experiment where first the communication pattern (i.e., the sequence of edges) is decided and then the adversary $\mathcal{A}$ chooses his initial sets and follows the above corruption paths (where edges are processed in the given order). For each such sequence of edges, $\mathcal{A}$ defines a probability distribution on the (active) edge set that is *fully* corrupted, namely, both its end-points are corrupted at the latest when they send any message in the protocol (and before they get a chance to erase it). Shifting the analysis from probabilistic party-corruption structures to probabilistic *edge*-corruption structures yields a simpler way to analyze the view of the experiment. Moreover, we provide a definition of what it means for an edge-corruption structure to be intolerable, which allows us to move back from edge to party corruptions.

Next, we define a *domination relation* which, intuitively, says that a probabilistic structure $\Pr_{\mathcal{A}_1^E}$ dominates another probabilistic structure $\Pr_{\mathcal{A}_2^E}$ on the same set of edges, if there exist a monotone probabilistic mapping F among sets of edges—i.e., a mapping from sets to their subsets—that transforms $\Pr_{\mathcal{A}_1^E}$ into $\Pr_{\mathcal{A}_2^E}$. Conceptually, for an adversary that corrupts according to $\Pr_{\mathcal{A}_1^E}$ (hereafter

referred to as a $\Pr_{\mathcal{A}_1^E}$-*adversary*), the use of F can be thought as "forgetting" some of the corrupted edges.[7] Hence, intuitively, an adversary who corrupts edge sets according to $\Pr_{\mathcal{A}_2^E}$ (or, equivalently, according to "$\Pr_{\mathcal{A}_1^E}$ with forget") is easier to simulate than a $\Pr_{\mathcal{A}_1^E}$-adversary, as if there is a simulator for the latter, we can apply the forget predicate F on the (simulated) set of corrupted edges to get a simulator for $\Pr_{\mathcal{A}_2^E}$. Thus, if $\Pr_{\mathcal{A}_2^E}$ is intolerable, then so is $\Pr_{\mathcal{A}_1^E}$.

Having such a domination relation in place, we next look for a simple probabilistic structure that is intolerable and can be dominated by the structure induced by our adversary $\mathcal{A}$. To this end, we prove intolerability of a special structure, where each edge set is sampled according to the following experiment: Let $\boldsymbol{E}$ be a collection of edge sets such that no $E \in \boldsymbol{E}$ can be derived as a union of the remaining sets; we choose to add each set from $\boldsymbol{E}$ to the corrupted-edge set independently with probability 1/2. The key feature of the resulting probabilistic corruption structure that enables us to prove intolerability and avoid missteps as in the above counterexample, is the independence of the above sampling strategy.

The final step, i.e., proving that the probabilistic edge-corruption structure induced by our adversary $\mathcal{A}$ dominates the above special structure, goes through a delicate combinatorial argument. We define a special graph traversing algorithm for the given edge sequence that yields a collection of potentially fully corruptible subsets of edges in this sequence, and prove that the maximal elements in this collection can be used to derive such a dominating probabilistic corruption structure.

The complete proof of our impossibility (stated in Theorem 4 below) can be found in [28].

Theorem 4. *Assume an adaptive passive adversary and that erasures are allowed. There exists no information theoretically secure protocol for computing the boolean OR function in the* $(2, n)$*-client/server model with message complexity* $m = o(n)$ *tolerating a* $(1, t)-$*adversary, where* $t = (1 - \sqrt{0.5} + \epsilon)n$ *for any constant* $\epsilon > 0$.

4.2 Security Without Erasures

We next turn to the case of adaptive corruptions (still for semi-honest adversaries) in a setting where parties do not erase any part of their state (and thus an adaptive adversary who corrupts any party gets to see the party's entire protocol view from the beginning of the protocol execution). This is another instance which demonstrates that requiring sublinear communication induces unexpected costs on the adversarial tolerance of MPC protocols.

In particular, when we do not restrict the communication complexity, then any $(1, t)$-adversary can be tolerated for information-theoretic MPC in the $(2, n)$-client/server model, as long as $t < n/2$ [4]. Instead, as we now show,

[7] Here, "forgetting" means removing the view of their end-points from the adversary's view.

when restricting to sublinear communication, there are functions that cannot be securely computed when any (arbitrary small) linear number of servers is corrupted (Theorem 5). If, on the other hand, we restrict the number of corruptions to be sublinear, a straightforward protocol computes any given function (Theorem 6).

The intuition behind the impossibility can be demonstrated by looking at protocol Π_{stat} from Sect. 3: An adaptive adversary can corrupt client c_1, wait for him to choose the servers in $\bar{\mathcal{S}}$, and then corrupt all of them rendering any protocol among them non-private. In fact, as we show below, this is not a problem of the protocol but an inherent limitation in the setting of adaptive security without erasures.

Specifically, the following theorem shows that if the adversary is adaptive and has the ability to corrupt as many servers as the protocols' message complexity, along with any one of the clients, then there are functions that cannot be privately computed. The basic idea is that such an adversary can wait until the end of the protocol, corrupt any of the two clients, say, c_i, and, by following the messages' paths, also corrupt all servers whose view is correlated to that of c_i. As we show, existence of a protocol tolerating such an adversary contradicts classical impossibility results in the MPC literature [4,35].

Theorem 5. *In the non-erasure model, there exists no information-theoretically secure protocol for computing the boolean OR function in the* $(2, n)$*-client/server model with message complexity* $m = o(n)$ *tolerating an adaptive* $(1, m + 1)$*-adversary.*

Proof. Assume towards contradiction that such a protocol Π exists. First we make the following observation: Let G denote the effective communication graph of the protocol defined as follows: $G = (V, E)$ is an undirected graph where the set V of nodes is the set of all parties, i.e., $V = \mathcal{S} \cup \{c_1, c_2\}$, and the set E of edge includes of pairs of parties that exchanged a message in the protocol execution; i.e., $E := \{(p_i, p_j) \in V^2$ s.t. p_i exchanged a message with p_j in the execution of $\Pi\}$.[8] By definition, the set $\bar{\mathcal{S}}$ of *active* parties is the set of nodes in G with degree $d > 0$. Let $\bar{\mathcal{S}}'$ denote the set of active parties that do not have a path to any of the two clients. (In other words, nodes in $\bar{\mathcal{S}}'$ do not belong in a connected component including c_1 or c_2.)

We observe that if a protocol is private against an adversary $\mathcal{A}$, then it remains private even if $\mathcal{A}$ gets access to the entire view of parties in $\bar{\mathcal{S}}'$ and of the inactive servers $\mathcal{S} \setminus \bar{\mathcal{S}}$. Indeed, the states of these parties are independent of the states of active parties and depend only on their internal randomness, hence they are perfectly simulatable.

Let $\mathcal{A}_1$ denote the adversary that attacks at the end of the protocol and chooses the parties A_1 to corrupt by the following greedy strategy: Initially $A_1 := \{c_1\}$, i.e., $\mathcal{A}_1$ always corrupts the first client. For $j = 1 \ldots, m$, $\mathcal{A}_1$ adds to A_1 all *servers* that are not already in A_1 and exchanged a message with

[8] Note that G is fully defined at the end of the protocol execution.

some party in A_1 during the protocol execution. (Observe that $\mathcal{A}_1$ does not corrupt the second client c_2.) Note that the corruption budget of the adversary is at least as big as the total message complexity, hence he is able to corrupt every active server (if they all happen to be in the same connected component as c_1). Symmetrically, we define the adversary $\mathcal{A}_2$ that starts with $A_2 = \{c_2\}$ and corrupts servers using the same greedy strategy. Clearly, $A_1 \cup A_2 = \bar{\mathcal{S}} \setminus \bar{\mathcal{S}}'$. Furthermore, as argued above, if Π can tolerate $\mathcal{A}_i$, then it can also tolerate $\mathcal{A}'_i$ which in addition to A_i learns the state of all servers in $\bar{\mathcal{S}}' \cup (\mathcal{S} \setminus \bar{\mathcal{S}})$; denote by A'_i the set of parties whose view $\mathcal{A}'_i$ learns. Clearly, $A'_1 \cup A'_2 = \mathcal{S}$, and thus, existence of such a Π contradicts the impossibility of computing the OR against non-Q^2 adversary structures [35]. □

Corollary 1. *In the non-erasure model, there exists no information theoretically secure protocol for computing the boolean OR function of the (two) clients' inputs with message complexity $m = o(n)$ tolerating an adaptive $(1, t)$-adversary, where $t = \epsilon n$ for some constant $\epsilon > 0$.*

For completeness, we show that if the adversary is restricted to a sublinear number t of corruptions, then there is a straightforward secure protocol with sublinear communication. Indeed, in this case we simply need to use Π_{stat}, with the modification that c_1 chooses $n' = 2t + 1$ servers to form a committee. Because $t = o(n)$, this committee is trivially of sublinear size, and because $n' > 2t$ a majority of the servers in the committee will be honest. Hence, the same argument as in Theorem 1 applies also here. This proves the following theorem; the proof uses the same structure as the proof of Theorem 1 and is therefore omitted.

Theorem 6. *Assuming $t = o(n)$, there exists an unconditionally secure (privately) protocol that computes any given 2-party function f in the $(2, n)$-client/server model in the presence of a passive adaptive $(1, t)$-adversary and communicates $o(n)|C_f|$ messages. The statement holds even when no erasures are allowed.*[9]

5 Sublinear Communication with Active (Static) Corruptions

Finally, we initiate the study of malicious adversaries in MPC with sublinear communication, restricting our attention to static security. Since the bound from Sect. 3 is necessary for semi-honest security, it is also necessary for malicious security (since a possible strategy of a malicious adversary is to play semi-honestly).

[9] A protocol that is secure when no erasures are allowed is also secure when erasures are allowed.

In this section we show that if $t < (1/2 - \epsilon)n$, then there exists a maliciously secure protocol for computing every two-party function *with abort*. To this end, we present a protocol which allows clients c_1 and c_2 to compute the 1-out-of-2 OT functionality $f_{\mathrm{OT}}((m_0, m_1), b) = (\perp, m_b)$ in the $(2, n)$-client/server model with sublinear communication complexity. As before, the completeness of OT ensures that this allows c_1 and c_2 to compute any function.

We remark that the impossibility result from Sect. 3 implies that no *fully* secure protocol (i.e., without abort) can tolerate a malicious $(1, t)$-adversary as above. As we argue below, the ability of the adversary to force an abort seems inherent in protocols with sublinear communication tolerating an active adversary with a linear number of corruptions. It is an interesting open question whether the impossibility of full security can be extended to malicious security with abort.

Towards designing a protocol for the malicious setting, one might be tempted to think that the semi-honest approach of one of the clients choosing a committee might work here as well. This is not the case, as this client might be corrupted (and malicious) and only pick servers that are also corrupted. Instead, here we use the following idea, inspired by the adaptive protocol with erasures (but without intermediaries): Every server independently decides with probability $p = \frac{\log^\delta n}{n}$ (based on his own local randomness) to volunteer in helping the clients by acting as an OT dealer. The choice of p is such that with overwhelming probability not too many honest servers (at most sublinear in n) volunteer. The clients then use the OT-combiner on the received pre-computed OT pairs to implement a secure OT. Note that this solution does not require any intermediaries as we have static corruptions.

But now we have a new problem to solve: The adversary might pretend to volunteer with more parties than the honest volunteers. (The adversary can do that since he is allowed a linear number of corruptions.) If the clients listen to all of them, then they will end up with precomputed OTs a majority of which is generated by the adversary. This is problematic since no OT combiner exists that will yield a secure OT protocol when the majority of the combined OTs is corrupted (cf. [34,47]).

We solve this problem as follows: We will have each of the clients abort during the OT pre-computation phase if he receives OT pairs from more than a (sublinear) number q of parties. By an appropriate choice of q we can ensure that if the adversary attempts to contact the clients with more corrupted parties than the honest volunteers, then with overwhelming probability he will provoke an abort. As a desirable added feature, this technique also protects against adversaries that try to increase the overall CC by sending more or longer messages. We note in passing that such an abort seems inevitable when trying to block such a message overflow by the adversary as the adversary is rushing and can make sure that his messages are always delivered before the honest parties' messages. The resulting protocol, $\Pi^{\mathrm{OT}}_{\mathrm{act}}$, is given below along with its security statement.

Protocol $\Pi_{\text{act}}^{\text{OT}}(\mathcal{C} = \{c_1, c_2\}, \mathcal{S} = \{s_1, \ldots, s_n\})$

1. Every server $s_i \in \mathcal{S}$ locally decides to become active with probability $p = \frac{\log^\delta n}{n}$ for a given (public) constant $\delta > 1$. Let $\bar{\mathcal{S}}$ denote the set of active servers.
2. Every $s_i \in \bar{\mathcal{S}}$ prepares an OT pair (m_i, r_i) and sends m_i to c_1 and r_i to c_2.
3. Each c_i, $i \in \{1, 2\}$, sends $\perp$ to c_{2-i} and aborts the protocol execution if c_i was contacted by more than $(1 - 16\epsilon^4)\log^\delta n$ parties in the previous step.
4. If c_i, $i \in \{1, 2\}$, received a $\perp$ from c_{2-i} in the previous step then he aborts.
5. The clients use the ℓ received OT pairs in a malicious $(\ell/2, \ell)$ OT-combiner [34] to obtain a secure OT protocol.

Theorem 7. *Protocol Π_{act}^{OT} unconditionally securely computes the function $f_{OT}((m_0, m_1), b) = (\perp, m_b)$ with abort in the $(2, n)$-client/server model in the presence of an active and static $(1, t)$-adversary with $t \leq (1/2 - \epsilon)n$, for any given $0 < \epsilon < 1/2$. Moreover, Π_{act}^{OT} communicates $O(\log^\delta(n))$ messages, for a given constant $\delta > 1$, except with negligible probability.*

Proof. Without loss of generality we can assume that adversary $\mathcal{A}$ corrupts $T = \lfloor(\frac{1}{2} - \epsilon)n\rfloor$ parties. Indeed, if the protocol can tolerate such an adversary then it can also tolerate any adversary corrupting $t \leq T$ parties.

For a given execution of $\Pi_{\text{act}}^{\text{OT}}$ let $\bar{\mathcal{S}}$ denote the set of servers that would become active if the adversary would behave semi-honestly (i.e., allow all corrupted parties to play according to the protocol). Then, each server $s \in \mathcal{S}$ is included in the set $\bar{\mathcal{S}}$ with probability $p = \frac{\log^\delta n}{n}$ independently of the other servers. Thus, by application of the Chernoff bound we get that for any constant $1 < \gamma < 0$:

$$\Pr[|\bar{\mathcal{S}}| \leq (1 - \gamma)\log^\delta n] < e^{-\frac{\gamma^2 \log^\delta n}{3}}.$$

For $\gamma = 4\epsilon^2$ he above equation implies that with overwhelming probability:

$$|\bar{\mathcal{S}}| > (1 - 4\epsilon^2)\log^\delta n. \tag{4}$$

Now let $C \subseteq \mathcal{S}$ denote the set of servers who are corrupted by the (static) adversary $\mathcal{A}$. (Recall that $\mathcal{A}$ corrupts $T = \lfloor(\frac{1}{2} - \epsilon)n\rfloor$ parties.) For each $s_i \in \bar{\mathcal{S}}$, let X_i denote the random variable which is 1 if $s_i \in C$ and 0 otherwise. Because the parties become OT dealers independently of the corruptions and the adversary corrupts T parties, $X_1, \ldots, X_{|\bar{\mathcal{S}}|}$ are i.i.d. random variables with $\Pr[X_i = 1] = T/n$. Thus, $X = \sum_{i=1}^{|\bar{\mathcal{S}}|} X_i = |\bar{\mathcal{S}} \cap C|$ with mean $\mu = \frac{|\bar{\mathcal{S}}|T}{n}$. By another application of the Chernoff bound we get that for any $0 < \epsilon_1 < 1$:

$$\Pr[|\bar{\mathcal{S}} \cap C| \geq (1 + \epsilon_1)\mu] < e^{-\frac{\epsilon_1^2 T}{3}}. \tag{5}$$

Hence, with overwhelming probability for $\epsilon_1 = 2\epsilon$:

$$|\bar{\mathcal{S}} \cap C| < (1 + \epsilon_1)\frac{T}{n}|\bar{\mathcal{S}}| \leq (1 + \epsilon_1)(\frac{1}{2} - \epsilon)|\bar{\mathcal{S}}| = (\frac{1}{2} - 2\epsilon^2)|\bar{\mathcal{S}}|.$$

Therefore, again with overwhelming probability the number h of honest parties that contact each of the parties as OT dealers is:

$$h = |\bar{\mathcal{S}} \setminus C| \geq \left(\frac{1}{2} + 2\epsilon^2\right) |\bar{\mathcal{S}}| \overset{(4)}{>} \left(\frac{1}{2} + 2\epsilon^2\right) (1 - 4\epsilon^2) \log^\delta n. \tag{6}$$

However, unless the honest client aborts, he accepts at most $\rho = (1 + \epsilon^2) \log^\delta n$ offers for dealers; therefore, the fraction of honest OT dealers among these ρ dealers is

$$\frac{h}{\rho} > \frac{(\frac{1}{2} + 2\epsilon^2)(1 - 4\epsilon^2)}{1 - 16\epsilon^4} = \frac{1}{2} \cdot \frac{(1 + 4\epsilon^2)(1 - 4\epsilon^2)}{1 - 16\epsilon^4} = \frac{1}{2}.$$

Thus, at least a 1/2 fraction of the OT vectors that an honest client receives is private and correct, in which case the security of protocol $\Pi^{\mathrm{OT}}_{\mathrm{act}}$ follows from the security of the underlying OT-combiner used in the last protocol step. □

Acknowledgements. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant #CNS-1523467. The second and third authors were supported in part by NSF-BSF grant 2015782 and BSF grant 2012366. The second author was additionally supported by ISF grant 1709/14, DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, 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 DARPA through the ARL under Contract W911NF-15-C-0205. The third author was additionally supported by NSF grant 1619348, DARPA, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. The views expressed are those of the authors and do not reflect the official policy or position of the DoD, the NSF, or the U.S. Government.

References

1. Beaver, D.: Precomputing oblivious transfer. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 97–109. Springer, Heidelberg (1995). doi:10.1007/3-540-44750-4_8
2. Beerliová-Trubíniová, Z., Hirt, M.: Efficient multi-party computation with dispute control. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 305–328. Springer, Heidelberg (2006). doi:10.1007/11681878_16
3. Beerliová-Trubíniová, Z., Hirt, M.: Perfectly-secure MPC with linear communication complexity. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 213–230. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78524-8_13
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: 20th ACM STOC, pp. 1–10. ACM Press, May 1988

5. Ben-Sasson, E., Fehr, S., Ostrovsky, R.: Near-linear unconditionally-secure multiparty computation with a dishonest minority. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 663–680. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_39
6. Boyle, E., Chung, K.-M., Pass, R.: Large-scale secure computation: multi-party computation for (parallel) RAM programs. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 742–762. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48000-7_36
7. Boyle, E., Goldwasser, S., Tessaro, S.: Communication locality in secure multiparty computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 356–376. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36594-2_21
8. Bracha, G.: An o(log n) expected rounds randomized byzantine generals protocol. J. ACM **34**(4), 910–920 (1987)
9. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**(1), 143–202 (2000)
10. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: 42nd FOCS, pp. 136–145. IEEE Computer Society Press, October 2001
11. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th ACM STOC, pp. 639–648. ACM Press, May 1996
12. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_2
13. Chandran, N., Chongchitmate, W., Garay, J.A., Goldwasser, S., Ostrovsky, R., Zikas, V.: The hidden graph model: communication locality and optimal resiliency with adaptive faults. In: Roughgarden, T. (ed.) ITCS 2015, pp. 153–162. ACM, January 2015
14. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: 20th ACM STOC, pp. 11–19. ACM Press, May 1988
15. Cohen, G., Damgård, I.B., Ishai, Y., Kölker, J., Miltersen, P.B., Raz, R., Rothblum, R.D.: Efficient multiparty protocols via log-depth threshold formulae. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 185–202. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40084-1_11
16. Cramer, R., Damgård, I., Ishai, Y.: Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 342–362. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30576-7_19
17. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 280–299. Springer, Heidelberg (2001). doi:10.1007/3-540-44987-6_18
18. 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 (2005). doi:10.1007/11535218_23
19. Damgård, I., Ishai, Y.: Scalable secure multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 501–520. Springer, Heidelberg (2006). doi:10.1007/11818175_30
20. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly secure multiparty computation and the computational overhead of cryptography. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 445–465. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_23

21. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000). doi:10.1007/3-540-44598-6_27
22. Damgård, I., Nielsen, J.B.: Universally composable efficient multiparty computation from threshold homomorphic encryption. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 247–264. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45146-4_15
23. Damgård, I., Nielsen, J.B.: Scalable and unconditionally secure multiparty computation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 572–590. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74143-5_32
24. Dani, V., King, V., Movahedi, M., Saia, J.: Brief announcement: breaking the o(nm) bit barrier, secure multiparty computation with a static adversary. In: Kowalski, D., Panconesi, A. (eds.) ACM Symposium on Principles of Distributed Computing, PODC 2012, Funchal, Madeira, Portugal, 16–18 July 2012, pp. 227–228. ACM (2012)
25. Dani, V., King, V., Movahedi, M., Saia, J.: Quorums quicken queries: efficient asynchronous secure multiparty computation. In: Chatterjee, M., Cao, J., Kothapalli, K., Rajsbaum, S. (eds.) ICDCN 2014. LNCS, vol. 8314, pp. 242–256. Springer, Heidelberg (2014). doi:10.1007/978-3-642-45249-9_16
26. Franklin, M., Haber, S.: Joint encryption and message-efficient secure computation. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 266–277. Springer, Heidelberg (1994). doi:10.1007/3-540-48329-2_23
27. Franklin, M.K., Yung, M.: Communication complexity of secure computation (extended abstract). In: 24th ACM STOC, pp. 699–710. ACM Press, May 1992
28. Garay, J., Ishai, Y., Ostrovsky, R., Zikas, V.: The price of low communication in secure multi-party computation. Cryptology ePrint Archive, Report 2017/520 (2017). http://eprint.iacr.org/2017/520
29. Genkin, D., Ishai, Y., Prabhakaran, M., Sahai, A., Tromer, E.: Circuits resilient to additive attacks with applications to secure computation. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 495–504. ACM Press, May/June 2014
30. Goldreich, O.: The Foundations of Cryptography - Volume 1, Basic Techniques. Cambridge University Press, Cambridge (2001)
31. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
32. 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
33. Harnik, D., Ishai, Y., Kushilevitz, E., Nielsen, J.B.: OT-combiners via secure computation. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 393–411. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78524-8_22
34. Harnik, D., Kilian, J., Naor, M., Reingold, O., Rosen, A.: On robust combiners for oblivious transfer and other primitives. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 96–113. Springer, Heidelberg (2005). doi:10.1007/11426639_6
35. Hirt, M., Maurer, U.M.: Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In: Burns, J.E., Attiya, H. (eds.) 16th ACM PODC, pp. 25–34. ACM, August 1997
36. Hirt, M., Maurer, U.M.: Player simulation and general adversary structures in perfect multiparty computation. J. Cryptol. **13**(1), 31–60 (2000)
37. Hirt, M., Maurer, U.: Robustness for free in unconditional multi-party computation. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 101–118. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_6

38. Hirt, M., Maurer, U., Przydatek, B.: Efficient secure multi-party computation. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 143–161. Springer, Heidelberg (2000). doi:10.1007/3-540-44448-3_12
39. Hirt, M., Nielsen, J.B.: Upper bounds on the communication complexity of optimally resilient cryptographic multiparty computation. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 79–99. Springer, Heidelberg (2005). doi:10.1007/11593447_5
40. Hirt, M., Zikas, V.: Adaptively secure broadcast. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 466–485. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_24
41. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963)
42. Ishai, Y., Ostrovsky, R., Zikas, V.: Secure multi-party computation with identifiable abort. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 369–386. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44381-1_21
43. 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 (2008). doi:10.1007/978-3-540-85174-5_32
44. Jakobsson, M., Juels, A.: Mix and match: secure function evaluation via ciphertexts. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 162–177. Springer, Heidelberg (2000). doi:10.1007/3-540-44448-3_13
45. Kilian, J.: Founding crytpography on oblivious transfer. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, pp. 20–31, New York, NY, USA. ACM Press (1988)
46. Lindell, Y., Pinkas, B.: A proof of security of Yao's protocol for two-party computation. J. Cryptol. **22**(2), 161–188 (2009)
47. Meier, R., Przydatek, B., Wullschleger, J.: Robuster combiners for oblivious transfer. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 404–418. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70936-7_22
48. Panconesi, A., Srinivasan, A.: Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. SIAM J. Comput. **26**(2), 350–368 (1997)
49. Rabin, M.O.: How to exchange secrets with oblivious transfer. Technical report TR-81, Aiken Computation Lab, Harvard University (1981)
50. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: 21st ACM STOC, pp. 73–85. ACM Press, May 1989
51. Shamir, A.: How to share a secret. Commun. Assoc. Comput. Mach. **22**(11), 612–613 (1979)
52. Yao, A.C.-C.: Protocols for secure computations (extended abstract). In: 23rd FOCS, pp. 160–164. IEEE Computer Society Press, November 1982