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, garay@yahoo-inc.com
[2] Department of Computer Science, Technion, yuvali@cs.technion.ac.il
[3] Department of Computer Science, UCLA, rafail@cs.ucla.edu
[4] Department of Computer Science, RPI, vzikas@cs.rpi.edu

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

We provide a complete investigation of security against semi-honest adversaries—static and adaptive, with and without erasures—and initiate the study of malicious adversaries. For semi-honest static adversaries, our bounds match (up to any constant fraction of corruptions) 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 a constant fraction of corruptions is intolerable, and—most 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 optimal up to an arbitrarily small constant factor. The latter optimality proof hinges on a new treatment of probabilistic adversary structures which may be of independent interest. In the case of active corruptions in this 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.

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 sees 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 [Yao82, GMW87] proved that under standard cryptographic assumption, e.g., the existence of enhanced trapdoor permutations (cf. [Gol04],) any multi-party function can be securely computed in the presence of a polynomially bounded semi-honest adversary corrupting arbitrary many parties. For the malicious case, [GMW87] proved that arbitrary many corruptions can be tolerated if we are willing to compromise only fairness—and achieve so-called *security with abort*—but an honest majority is required if we insists on achieving also fairness.

In the information-theoretic (IT) model—where there are no restrictions on the adversary's computational power—the situation is different. Ben-Or, Goldwasser, and Wigderson [BGW88] and independently Chaum, Crépeau, and Damgård [CCD88b] proved that IT security is possible if

and only if $t < n/3$ parties are actively corrupted, (respectively, if and only if $t < n/2$ are passively corrupted.) The solutions of [BGW88] are perfectly secure, i.e., there is a zero-error probability. Rabin and Ben-Or [RB89] 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 all corrupted parties 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 not relying on computational assumptions, information theoretic protocols typically enjoy strong composability guarantees. Concretely, the above conditions for the IT setting allow for universally-composable (UC) secure protocols [Can01]. This is known to be impossible for the weaker computational bounds in the *plain model*—i.e., without assuming access to a trusted setup functionality such as a common reference string (CRS) [CF01]. Given the above advantages of IT protocols, it is natural to investigate ways to obtain IT-secure protocols for arbitrary many corruptions.

It is well known that assuming a strong setup such as *oblivious transfer* (OT) [Rab81], we can construct IT secure protocols tolerating arbitrary many corruptions both in the semi-honest [GMW87] and in the malicious setting [Kil88a, IPS08]. However, these solutions require trusting a (centralised party that serves as) an OT functionality.

An alternative approach is for the parties to use help from other servers in a network the have access to, e.g., the Internet. This naturally leads to the formulation of the problem in the so-called *client-server* model [CDI05, DI05, DI06, HM00]. This model refines the standard MPC model by separating parties into *clients*, who wish to perform some computation and provide the inputs and receive outputs to 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.

The direct approach to obtain security in the client/server model is to have the clients share their input to *all* the servers—in the following we denote by n the number of servers—who perform the computation on these inputs return to the clients their respective outputs. Using [GMW87, BGW88, CCD88b, RB89] this yields a protocol tolerating $t < n/2$ semi-honest corrupted servers or, for the malicious setting, $t < n/2$ corrupted servers if broadcast among them is available, and $t < n/3$, otherwise. (Recall that the above bounds are in additions to arbitrary many corruptions of clients).

Despite its simplicity, the above approach incurs a high overhead in communication when the number of clients is small in comparison with the number of servers—this is usually the case in natural 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 optimally resilient IT protocols in the client/server model whose communication is sublinear in the number n of servers. As we prove, this low-communication requirement comes at a cost, i.e., it induces a different landscape of feasibility bounds.

Related Literature. The literature of communication complexity (CC) of MPC is vast. To put out results in perspective, in the following we discuss some of the most relevant literature on

IT MPC with low communication complexity. For notation 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 logarithmic in n .

The CC of the original protocols from the 80's was polynomial (in the best case quadratic) in n, in particular, $poly(n) \cdot |C|$ where $|C|$ denotes the size of the circuit C that computes the given function. A long line of work ensued (e.g., [FH94, FY92, HMP00, HM01, JJ00, CDN01, DN03, HN05, BH06, BH08, IPS08, DKMS12, DKMS14, BCP15]) has reduced this complexity down to linear in the size of the party set by shifting the dependency on different parameters.

Concretely in the IT setting, Damgård and Nielsen [DN07] achieve a CC of $O(n|C| + n^2)$ messages—i.e., their CC scales in a linear fashion with the size $|C|$ of the circuit C for computing the given function. Their protocol tolerates $t < n/2$ semi-honest corruptions. In the malicious setting, they provided a protocol for tolerating $t < n/3$ corruptions with a CC of $O(n|C| + d \cdot n^2) + poly(n)$ messages, where d is the multiplicative depth of the circuit C. Beerliova and Hirt [BH08] extended this result to perfect security and achieve CC of $O(n|C| + d \cdot n^2 + n^3)$. Ben-Sasson, Fehr, and Ostrovsky [BFO12] achieved CC $O(n|C| + d \cdot n^2) + poly(n)$ messages against $t < n/2$ active corruptions which was brought down to $O(n|C| + n^2)$ by Genkin et al. [GIP+14]. Note that with the exception of the maliciously secure protocol from [DN07], all the above works tolerate a number of corruptions which is tight even for the case where no bound on the communication complexity is considered.

The first to look at achieving a scalability factor that is sublinear in the number of parties were Damgård and Ishai [DI06] but for the computational setting. In the information-theoretic setting Damgård, Ishai, and Krøigaard [DIK10] devised a protocol with CC $O(\log n \cdot |C|) + poly(n, d)$ messages tolerating an asymptotically optimal bound of $t < (1/2 - \epsilon)n$ malicious corruptions for an arbitrary small constant $\epsilon > 0$.

We point out that all the above results incur polynomial, in n, additive factors on 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, even worse, messages) which is polynomial in n. Instead, in this work we are interested in achieving overall communication (bit) complexity of $o(n)|C|$ without such additive (polynomial or even linear in n) factors.

Finally, a different line of work [BGT13, CCG+15, BCP15] focuses on reducing *communication locality* of MPC protocols. This corresponds to the maximum number of neighbors/parties that any party communicates with *directly*, i.e., via a bilateral channel, throughout the protocol. Although these works achieve a sublinear (in the number n of parties) communication locality, they communicate an at least polynomial number of bits—similarly to the scalable MPC literature discussed above—therefore failing to achieve our goal of sublinear bit complexity.

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.[5] 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:

[5] 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.

- 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" [Bra87] to help the clients perform their computation, is secure and has sublinear message complexity against $t < (1/2 - \epsilon)n$ corrupted servers, for any given constant $0 < \epsilon < 1/2$. Further, we prove that this bound is tight. Thus, up to an arbitrary 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 counter-intuitive complication in providing optimally resilient protocols. Concretely, 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[6] 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.

  Specifically, 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 can be at most 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). Existence of such an adversary contradicts classical MPC impossibility results [HM97], that prove that if there is a two-set partition of the party-set and the adversary might corrupt either of the sets (this is the so called Q^2 condition in [HM97]) 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 assumed. We prove that any adversary which corrupts at most $t < (1 - \sqrt{0.5} - \epsilon)n$ of the parties, for any constant $0 < \epsilon < 1 - \sqrt{0.5}$, can be tolerated, a bound which is asymptotically 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 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

[6] As opposed to requiring the use of more complex cryptographic tools such as non-committing encryption [CFGN96, DN00] as in the non-erasure setting.

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 fair abort [GMW87, IOZ14] 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. Proving wether or not such an abort is indeed necessary and/or extending our treatment to adaptive active corruptions is left as an open question.

We 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).

Organization of the paper. In Section 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 Section 4 is dedicated to semi-honest adaptive corruptions, with erasures (Section 4.1) and without erasures (Section 4.2). Finally, Section 5 includes our feasibility result for malicious (static) adversaries.

2 Model, Definitions and Building Blocks

We consider a set of $n+2$ 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 unconditional secure multi-party computation (MPC) protocols [BGW88, CCD88a]. We call this model the $(2, n)$-*client/server model*.

The parties wish to compute a given two-party function 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. [LP09].) 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 see 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* [Can00, HZ10], 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. More concretely, every round is partitioned in 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, simultaneously, as an atomic operation (cf. [HZ10] for a related discussion about atomicity of sending messages).

The communication complexity of a protocol is the number of bits that are sent or received by honest parties during a protocol execution. Throughout this work we will consider protocols in which honest (or semi-honest) parties send at most a sublinear (in n) number of messages in the protocol, i.e., the communication complexity is $o(n)$. Furthermore, we will only consider information-theoretic security (see below).

Simulation-based security. We will use the standard simulation-based definition of security from [Can00]. 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. More concretely, 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 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 [Can00] 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* [Gol01], 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:

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

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 or 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) [Rab81] is a two-party functionality between a *sender* and a *receiver*. In its most common variant called 1-out-of-2-OT,[7] 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, Micali, and Wigderson [GMW87,Kil88b], OT is a complete primitive for two-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.* [IPS08].

Beaver [Bea95] showed how OT can be precomputed, 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 his part of it. We will refer to such a correlated randomness setup (R_s, R_r) where the sender receives R_s and the receiver R_r as and *OT pair*. The size of each component in such an OT pair is the same as (or linear in) the size of the 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 m OT protocols out of which ℓ might be insecure, and combine them into a secure

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

OT protocol (e.g., [HKN+05, MPW07, HIKN08]). Such a combiner with linear rate (i.e., where the total communication of the combiner is linear in the total communication of the OT protocol) exists 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 precomputed 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 prove that 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.[8]

Concretely, in the static semi-honest case the following "folklore" protocol based on the approach of selecting a random committee [Bra87] 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 this choice. These servers are given as input secret sharings of the client's inputs, and are requested to act as servers in a standard MPC protocol that is secure in the presence of an honest majority, for example, the semi-honest MPC protocol by Ben-Or, Goldwasser and Wigderson [BGW88], 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. The protocol outlined above, denoted Π_{stat}, is specified in more detail below and its security is stated in Theorem 1. The proof is simple and follows the above idea; it can be found in Appendix A.

Protocol $\Pi_{\text{stat}}(\mathcal{C} = \{c_1, c_2\}, \mathcal{S} = \{s_1, \ldots, s_n\}, f)$

1. Client c_1 chooses a subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$ of $\log^\delta n$ servers (i.e., $|\bar{\mathcal{S}}| = \log^\delta n$) for some constant $\delta > 1$ uniformly at random. Client c_1 sends (the identities of) $\bar{\mathcal{S}}$ to c_2 and to every party in $\bar{\mathcal{S}}$.
2. Each c_i secret-shares his input x_i to the parties in $\bar{\mathcal{S}}$ by means of a $\frac{|\bar{\mathcal{S}}|}{2}$-out-of-$|\bar{\mathcal{S}}|$ secret-sharing scheme (e.g., using Shamir's polynomial secret sharing [Sha79] with degree $t = \frac{\lfloor |\bar{\mathcal{S}}| \rfloor}{2}$).
3. The servers in $\bar{\mathcal{S}}$ invoke an MPC protocol which is unconditionally secure for an honest majority (e.g., the BGW protocol) to compute the function f' that on inputs the shares distributed by the clients performs the following computation: It privately reconstructs x_1 and x_2, evaluates f on these inputs (i.e., computes $y = f(x_1, x_2)$), and outputs y towards a default server $s \in \bar{\mathcal{S}}$ (the one with the smallest index, for example).
4. s sends y to both clients who output it and halt.

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 $t < (1/2 - \epsilon)n$,*

[8] Wlog we can assume that the semi-honest adversary just outputs his entire view [Can00]; hence semi-honest adversaries only differ in the set of parties they corrupt.

for any given constant $0 < \epsilon < 1/2$. *Moreover,* Π_{stat} *communicates* $O(\log^{\delta'}(n))$ *messages, for a constant* $\delta' > 1$.

Next, we prove that Theorem 1 is tight. The proof idea is as follows: If the adversary corrupts almost a majority of the parties, no matter which subset of the servers is actually activated (i.e., sends or receives a message) in the protocol[9], 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 [HM97] which implies that an adversary who can corrupt a set and its complement—or supersets thereof—is intolerable for the OR function.

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* $\delta = O(1)$.

Proof. Assume towards contradiction that a protocol Π as in the above theorem exists. Let $\bar{\mathcal{S}}$ denote the set of *active* servers at the end of the protocol execution (i.e., the set of servers that send or receive a message during the protocol execution). By the message-complexity assumption, $|\bar{\mathcal{S}}| \leq m = o(n)$ since each server in $\bar{\mathcal{S}}$ has to participate in at least one message exchange.

Consider the $(1, t)$-adversary $\mathcal{A}_1$ that corrupts one of the two client randomly (i.e., each with probability $1/2$) and additionally corrupts t servers as follows: First he picks a subset $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| = \delta$ parties uniformly at random which he does not corrupt, and from the remaining $n - \delta$ parties he corrupts a random subset $A_1 \subseteq (\mathcal{S} \backslash \mathcal{S}')$ with $|A_1| = \lceil n/2 - \delta \rceil$. Consider now the adversary $\mathcal{A}_2$ that emulates a copy of $\mathcal{A}_1$ but corrupts the client that $\mathcal{A}_1$ does not corrupt, and also corrupts all the servers in $\mathcal{S} \backslash \mathcal{S}'$ that $\mathcal{A}_1$ does not corrupt ($\mathcal{A}_2$ also leaves out the servers in $\mathcal{S}'$). Let A_2 denote the set of parties corrupted by $\mathcal{A}_2$. Clearly, if $\mathcal{S}' \cap \bar{\mathcal{S}} = \emptyset$ then $A_1 \cup A_2 \subseteq \bar{\mathcal{S}}$. But with probability $p \geq 1 - \frac{\delta m}{n}$ (which is not negligible, since $\delta = O(1)$ and $m = o(n)$) we do have $\mathcal{S}' \cap \bar{\mathcal{S}} = \emptyset$. This holds because each of the parties in $\mathcal{S}'$ is in $\bar{\mathcal{S}}$ with probability at most $\frac{|\bar{\mathcal{S}}|}{n} \leq \frac{m}{n}$ (recall that $\mathcal{S}'$and $\bar{\mathcal{S}}$ are chosen uniformly and independently.) Thus, a protocol which is secure against both $\mathcal{A}_1$ and $\mathcal{A}_2$ would have to be private against A_1 or A_2 being corrupted with $\bar{\mathcal{S}} \subseteq A_1 \cup A_2$.

Next, we observe that if a protocol is private against some adversary, then it remains private even if the adversary gets access to the entire view of the inactive servers. Indeed, the states of these servers are independent of the states of active parties and depend only on their internal randomness, hence they are perfectly simulatable. Thus, 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 $(\mathcal{S} \setminus \bar{\mathcal{S}})$; denote by A_i' the (random variable corresponding to the) set of parties that $\mathcal{A}_i'$ learns their view. The above analysis implies that $A_1' \cup A_2' = \mathcal{S}$ with noticeable probability. Hence, if Π tolerates adversary $\mathcal{A}$ it also tolerates an adversary choosing to corrupt between A_1' and A_2'—where with noticeable probability $A_1' \cup A_2' = \mathcal{S}$—and also corrupting any one of the two clients; existence of such a Π contradicts the impossibility of computing the OR against non-Q^2 adversary structures [HM97]. □

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 pro-

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

tect 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 parties' states are allowed, which prominently demonstrates that sublinear communication comes at an unexpected cost in the number of corruptions that can be tolerated. 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 (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 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 $f_{OT}((m_0, m_1), b) = (\perp, m_b)$ 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., [HKN$^+$05]) on the received pre-computed 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 Π^{OT}_{adap} is shown below, followed by the statement of its security.

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

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 $((m_i, r_i), \texttt{otid}_i)$, where $\texttt{otid}_i \in \{0,1\}^{\log^\delta n}$ is a uniformly chosen identifier.
2. Every $s_i \in \bar{\mathcal{S}}_1$ choses a relayer $s_{ij} \in \mathcal{S}$ uniformly at random and sends $(r_i, \texttt{otid}_i)$ to s_{ij}. Denote by $\bar{\mathcal{S}}_2 = \{s_{ij} | s_i \in \bar{\mathcal{S}}\}$ the set of all relayers (i.e., intermediaries).
3. Every $s_i \in \bar{\mathcal{S}}_1$ erases r_i and the randomness used to select s_{ij}.
4. Every $s_i \in \bar{\mathcal{S}}_1$ sends $(m_i, \texttt{otid}_i)$ to c_1 and every $s_{ij} \in \bar{\mathcal{S}}_2$ sends $(r_i, \texttt{otid}_i)$ to c_2.
5. Every $s_i \in \bar{\mathcal{S}}_1$ and every $s_{ij} \in \bar{\mathcal{S}}_2$ erase its entire internal state.
6. The clients c_1 and c_2 use the OT pairs with matching $\texttt{otid}$'s within a (semi-honest) $(n/2, n)$ OT-combiner [HKN+05] 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 $0 < \gamma < 1$:

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

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 $0 < \gamma' < 1$:

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

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 has 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 $s_{ij} \in \bar{\mathcal{S}}_2$ communicate 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 subset of $\bar{\mathcal{S}}'_1$ 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" with respect 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 $s \in \bar{\mathcal{S}}_1$ 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. As we will show below, 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 $\bar{\mathcal{S}}_1$. To this direction, we first prove in the following lemma 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 for any constant $1 > \epsilon' > 0$ and for big 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 [PS97] (see Theorem 8 in Appendix B.) 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 a relayer in the first step of the protocol.

$$\begin{aligned} \Pr[X_i = 1] &= \Pr[(s_{ij} \in \bar{\mathcal{S}}_1) \cup (\exists s_k \in \bar{\mathcal{S}}_1 \text{ s.t. } s_{kj} \in \{s_i, s_{ij}\})] \\ &\leq \Pr[s_{ij} \in \bar{\mathcal{S}}_1] + \Pr[\exists s_k \in \bar{\mathcal{S}}_1 \text{ s.t. } s_{kj} = s_i] + \Pr[\exists s_k \in \bar{\mathcal{S}}_1 \text{ s.t. } s_{kj} = s_{ij}] \\ &\leq 3\frac{|\bar{\mathcal{S}}_1|}{n} \end{aligned} \tag{2}$$

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 a

relayer. But from Equation 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 mush 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 a direct application of the generalised Chernoff bound (Theorem 8) 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. □

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. To this direction, 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 $|C' \cap \bar{\mathcal{S}}_1| < \frac{1}{2}|\bar{\mathcal{S}}_1|$, 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 of 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 :

$$\Pr[X_i = 1] \leq \frac{2\lambda n}{(1-\epsilon')n} - \left(\frac{\lambda n - |\bar{\mathcal{S}}_1'|}{n - |\bar{\mathcal{S}}_1'|}\right)^2 + \mu \tag{3}$$

Moreover,

$$\begin{aligned}\left(\frac{\lambda n - |\bar{\mathcal{S}}_1'|}{n - |\bar{\mathcal{S}}_1'|}\right)^2 &\geq \left(\frac{\lambda n - |\bar{\mathcal{S}}_1'|}{n}\right)^2 = \left(\lambda - \frac{|\bar{\mathcal{S}}_1'|}{n}\right)^2 \\ &\geq \lambda^2 - \frac{2\lambda|\bar{\mathcal{S}}_1'|}{n}\end{aligned} \tag{4}$$

But because, from Equation 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 the following holds: $\frac{2\lambda|\bar{\mathcal{S}}_1'|}{n} + \mu' < \epsilon_1$. thus combining Equations 3 and 4 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 Theorem 8 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 $0 < \epsilon < 1-\sqrt{0.5}$. To this direction, 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 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 [HM00, CDI+13] 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 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.[10] 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 in $\boldsymbol{E}$ 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 miss-steps as in the above counterexample, is the independence assumption in the above sampling game.

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 Appendix C.

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 a 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 view (and thus an adaptive adversary that corrupts any parties gets to see the parties' 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 protocols' adversarial tolerance.

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

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 $t < n/2$ [BGW88]. Furthermore, exact protocols are typically tougher to come up with in the case of no erasures. Instead, as we now show, 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, there is a straightforward protocol that computes any given function (Theorem 6).

The intuition behind the impossibility can be demonstrated by looking at protocol Π_{stat} from Section 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 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 relevant for the computation. As we show, existence of a protocol tolerating such an adversary contradicts classical impossibility results in the MPC literature [BGW88, HM97].

Theorem 5. *In the non-erasure model, there exists no information theoretically secure protocol for computing the 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., $E := \{(p_i, p_j) \in V^2 \text{ s.t. } p_i \text{ exchanged a message with } p_j \text{ in the execution of } \Pi\}$. [11] 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 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 even 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

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

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 that $\mathcal{A}'_i$ learns their view. Clearly, $A'_1 \cup A'_2 = \mathcal{S}$, thus existence of such a Π contradicts the impossibility of computing the OR against non-Q^2 adversary structures [HM97]. □

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 corrupted servers, then there is a straightforward way to tolerate this adversary by a sublinear communication protocol. 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 $o(n)$ messages. The statement holds even when no erasures are allowed.*[12]

5 Sublinear Communication with Active (Static) Corruptions

Next, we initiate the study of malicious adversaries on the MPC setting with sublinear communication, restricting our attention to static security. Since the bound from Section 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). 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* against a $(1,t)$-adversary. To this end, we present a protocol which allows clients c_1 and c_2 to compute the 1-out-of-2 OT functionality $f_{OT}((m_0, m_1), b) = (\perp, m_b)$ 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 Section 3 implies that no *fully* secure protocol (i.e., without abort) can tolerate a $(1,t)$-adversary as above. As we argue below, the ability of the adversary to force an abort seems inherent in protocols that achieve sublinear communication against an active adversary with a linear number of corruptions. Thus, it is an interesting open question whether the semi-honest impossibility can be extended to the case of security with abort.

Before presenting our protocol for this setting, we discuss a subtle issue with sub-linear communication complexity when a constant fraction of the parties might be corrupted by a malicious adversary. Concretely, such a malicious adversary is able to corrupt a linear number of parties and can therefore send a linear number of bits to honest parties. Hence, strictly speaking, a linear number of messages is sent through the point-to-point channels. However, in our protocol we will have all honest parties block their communication interfaces if they receive too many messages (and

[12] Observe that a protocol that is secure when no erasures are allowed is trivially also secure when erasures are allowed.

in any case, before they exceed their sublinear budget). This is similar to how sublinear communication locality is achieved in [BGT13, CCG+15] in the presence of a linear number of corrupted parties. In particular, this will ensure that the total number of messages that *honest* parties send or receive in the protocol is sublinear.

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 volunteer (at most sublinear in n). 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 two problems to solve. First, the adversary might pretend to be volunteering with more than a sublinear number of parties (since he is allowed a linear number of corruptions). If the clients listen to all of them then we will end up with a higher than sublinear communication complexity. Second, even if the adversary only volunteers with a sublinear number of corrupted servers, it might still be that the majority of the volunteers is corrupted, and no OT combiner exists that will yield a secure OT protocol when the majority of the combined OTs is corrupted (cf. [HKN+05, MPW07]).

Both these problems are solved 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 (sub-linear) number q of parties. Most importantly, 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. 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, Π_{act}, is given below along with its security statement.

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

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 parties that become active.
2. Every $s_i \in \bar{\mathcal{S}}$ prepares $\lambda = \mathsf{poly}(k)$ OT pairs $(m_{i1}, r_{i1}), \ldots, (m_{i\lambda}, r_{i\lambda})$ and sends the vectors $(m_{i1}, \ldots, m_{i\lambda})$ and $(r_{i1}, \ldots, r_{i\lambda})$ to clients c_1 and c_2, respectively.
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 $(1 + \epsilon^2) \log^\delta n$ 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 parties use the OT pairs with a malicious $(n/2, n)$ OT-combiner [HKN+05] 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 than it can also tolerate any adversary corrupting $t \leq T$ parties.
For a given execution of Π_{act}^{OT} let $\bar{\mathcal{S}}$ denote the set of servers that would become corrupted if the adversary would be passive (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 $p = \frac{\log^{\delta} n}{n}$ independent 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 $\gamma = \epsilon^2$ he above equation implies that with overwhelming probability:

$$|\bar{\mathcal{S}}| > (1 - \epsilon^2) \log^{\delta} n. \tag{5}$$

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 indicator 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 $0 < \epsilon_1 < 1$:

$$\Pr[|\bar{\mathcal{S}} \cap C| \geq (1 + \epsilon_1)\mu] < e^{-\frac{\epsilon_1^2 T}{3}}, \tag{6}$$

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

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

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

$$h = |\bar{\mathcal{S}} \setminus C| \geq \left(\frac{1}{2} + \epsilon^2\right)|\bar{\mathcal{S}}| \overset{(5)}{>} \left(\frac{1}{2} + \epsilon^2\right)(1 - \epsilon^2) \log^{\delta} n \tag{7}$$

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

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

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 Π_{act}^{OT} follows from the security of the underlying OT-combiner used in the last protocol step. □

References

BCP15. Elette Boyle, Kai-Min Chung, and Rafael Pass. Large-scale secure computation: Multi-party computation for (parallel) RAM programs. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 742–762. Springer, Heidelberg, August 2015.

Bea95. Donald Beaver. Precomputing oblivious transfer. In Don Coppersmith, editor, *Advances in Cryptology — CRYPTO' 95: 15th Annual International Cryptology Conference Santa Barbara, California, USA, August 27–31, 1995 Proceedings*, pages 97–109, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

BFO12. Eli Ben-Sasson, Serge Fehr, and Rafail Ostrovsky. Near-linear unconditionally-secure multiparty computation with a dishonest minority. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 663–680. Springer, Heidelberg, August 2012.

BGT13. Elette Boyle, Shafi Goldwasser, and Stefano Tessaro. Communication locality in secure multi-party computation - how to run sublinear algorithms in a distributed setting. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 356–376. Springer, Heidelberg, March 2013.

BGW88. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988.

BH06. Zuzana Beerliová-Trubíniová and Martin Hirt. Efficient multi-party computation with dispute control. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 305–328. Springer, Heidelberg, March 2006.

BH08. Zuzana Beerliová-Trubíniová and Martin Hirt. Perfectly-secure MPC with linear communication complexity. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 213–230. Springer, Heidelberg, March 2008.

Bra87. Gabriel Bracha. An o(log n) expected rounds randomized byzantine generals protocol. *J. ACM*, 34(4):910–920, October 1987.

Can00. Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

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

CCD88a. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (abstract) (informal contribution). In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, page 462. Springer, Heidelberg, August 1988.

CCD88b. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *20th ACM STOC*, pages 11–19. ACM Press, May 1988.

CCG+15. Nishanth Chandran, Wutichai Chongchitmate, Juan A. Garay, Shafi Goldwasser, Rafail Ostrovsky, and Vassilis Zikas. The hidden graph model: Communication locality and optimal resiliency with adaptive faults. In Tim Roughgarden, editor, *ITCS 2015*, pages 153–162. ACM, January 2015.

CDI05. Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 342–362. Springer, Heidelberg, February 2005.

CDI+13. Gil Cohen, Ivan Bjerre Damgård, Yuval Ishai, Jonas Kölker, Peter Bro Miltersen, Ran Raz, and Ron D. Rothblum. Efficient multiparty protocols via log-depth threshold formulae - (extended abstract). In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 185–202. Springer, 2013.

CDN01. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 280–299. Springer, Heidelberg, May 2001.

CF01. Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–40. Springer, Heidelberg, August 2001.

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

DI05. Ivan Damgård and Yuval Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 378–394. Springer, Heidelberg, August 2005.

DI06. Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 501–520. Springer, Heidelberg, August 2006.

DIK10. Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 445–465. Springer, Heidelberg, May 2010.

DKMS12. Varsha Dani, Valerie King, Mahnush Movahedi, and Jared Saia. Brief announcement: breaking the o(nm) bit barrier, secure multiparty computation with a static adversary. In Darek Kowalski and Alessandro Panconesi, editors, *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*, pages 227–228. ACM, 2012.

DKMS14. Varsha Dani, Valerie King, Mahnush Movahedi, and Jared Saia. Quorums quicken queries: Efficient asynchronous secure multiparty computation. In Mainak Chatterjee, Jian-Nong Cao, Kishore Kothapalli, and Sergio Rajsbaum, editors, *Distributed Computing and Networking - 15th International Conference, ICDCN 2014, Coimbatore, India, January 4-7, 2014. Proceedings*, volume 8314 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2014.

DN00. Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 432–450. Springer, Heidelberg, August 2000.

DN03. Ivan Damgård and Jesper Buus Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 247–264. Springer, Heidelberg, August 2003.

DN07. Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 572–590. Springer, Heidelberg, August 2007.

FH94. Matthew K. Franklin and Stuart Haber. Joint encryption and message-efficient secure computation. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 266–277. Springer, Heidelberg, August 1994.

FY92. Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *24th ACM STOC*, pages 699–710. ACM Press, May 1992.

GIP+14. Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In David B. Shmoys, editor, *46th ACM STOC*, pages 495–504. ACM Press, May / June 2014.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

Gol01. Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

Gol04. Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

HIKN08. Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. Ot-combiners via secure computation. In Ran Canetti, editor, *Theory of Cryptography: Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. Proceedings*, pages 393–411, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

HKN+05. Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 96–113. Springer, Heidelberg, May 2005.

HM97. Martin Hirt and Ueli M. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In James E. Burns and Hagit Attiya, editors, *16th ACM PODC*, pages 25–34. ACM, August 1997.

HM00. Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000.

HM01. Martin Hirt and Ueli M. Maurer. Robustness for free in unconditional multi-party computation. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 101–118. Springer, Heidelberg, August 2001.

HMP00. Martin Hirt, Ueli M. Maurer, and Bartosz Przydatek. Efficient secure multi-party computation. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 143–161. Springer, Heidelberg, December 2000.

HN05. Martin Hirt and Jesper Buus Nielsen. Upper bounds on the communication complexity of optimally resilient cryptographic multiparty computation. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 79–99. Springer, Heidelberg, December 2005.

Hoe63. Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):pp. 13–30, 1963.

HZ10. Martin Hirt and Vassilis Zikas. Adaptively secure broadcast. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 466–485. Springer, Heidelberg, May 2010.

IOZ14. Yuval Ishai, Rafail Ostrovsky, and Vassilis Zikas. Secure multi-party computation with identifiable abort. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 369–386. Springer, Heidelberg, August 2014.

IPS08. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, Heidelberg, August 2008.

JJ00. Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 162–177. Springer, Heidelberg, December 2000.

Kil88a. Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

Kil88b. Joe Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31, New York, NY, USA, 1988. ACM.

LP09. Yehuda Lindell and Benny Pinkas. A proof of security of Yao's protocol for two-party computation. *Journal of Cryptology*, 22(2):161–188, April 2009.

MPW07. Remo Meier, Bartosz Przydatek, and Jürg Wullschleger. Robuster combiners for oblivious transfer. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 404–418. Springer, Heidelberg, February 2007.

PS97. Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997.

Rab81. Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981.

RB89. Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *21st ACM STOC*, pages 73–85. ACM Press, May 1989.

Sha79. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

Yao82. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982.

A Semi-Honest Static Adversaries

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 $O(\log^{\delta'}(n))$ messages, for a constant $\delta' > 1$.*

Proof. The bound on the communication complexity follows immediately from the fact that there are $\log^{\delta} n$ active servers that send/receive messages in the computation; denote them by $\bar{\mathcal{S}}$. The only messages exchanged between the clients and the servers are the inputs and outputs and the indexes of the parties in $\bar{\mathcal{S}}$; on top of that, the servers exchange their messages in protocol BGW which are polynomially many in $|\bar{\mathcal{S}}| = \log^{\delta} n$. Thus, overall, the total number of exchanged messages is polynomial in $|\bar{\mathcal{S}}| = \log^{\delta} n$.

To complete the proof, we need to show that with overwhelming probability, less than half of the active servers are corrupted. Indeed, when this is the case, the secret sharings in Step 2 reveal no information to the adversary, as does the execution of the [BGW88] protocol in the Step 3. Thus, the computation is private.

The fact that less than half of the active servers are corrupted follows as a corollary, by a direct application of Hoeffding's inequality along the lines of [CCG$^+$15, Lemma 10].

Corollary 2. *Assume that set $\bar{\mathcal{S}}$ is chosen as in Π_{stat} and the adversary corrupts $t < (1/2 - \epsilon)n$. Then with overwhelming probability, the adversary corrupts less than $|\bar{\mathcal{S}}|/2$ parties in $\bar{\mathcal{S}}$.*

Proof. The corollary follows immediately by Lemma [CCG$^+$15, Lemma 10] (cf. Appendix A) by setting $V = \mathcal{S}, C = \bar{\mathcal{S}}$, and U be the set of corrupted parties. □

□

Lemma 3. (Hoeffding's Inequality [Hoe63]) *Let $S = \{x_1, \ldots, x_N\}$ be a finite set of real numbers with $a = \min_i x_i$ and $b = \max_i x_i$. Let $X_1, \ldots, X_n$ be a random sample drawn from S without replacement. Let $\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ and $\mu = \frac{\sum_{i=1}^{N} x_i}{N} = E[X_j]$. Then for all $\delta > 0$, $\Pr[\overline{X} - \mu \geq \delta] \leq e^{-\frac{2n\delta^2}{(b-a)^2}}$.*

Lemma 4 ([CCG$^+$15, Lemma 10]). *Let $V = [n]$ and $C \subseteq V$, $|C| = m$, be a subset chosen uniformly at random. Let $0 < q < 1$ be a constant and $U \subseteq V$, $|U| = qn$, be a subset chosen independently of C. Then, for all $0 < \delta < 1 - q$, $|C \setminus U| > (1 - q - \delta)m$ except with probability $e^{-2m\delta^2}$. In particular, for $m = \log^{1+\epsilon'} n$, $|C \setminus U| > \left(\frac{1-q}{2}\right) m$ except with negligible probability. Furthermore, for $q = \frac{1}{2} - \epsilon$, $|C \setminus U| > \frac{1}{2}m$ except with negligible probability.*

Proof. Let $S = \{x_1, \ldots, x_n\}$ where $x_i = 1$ if $i \in U$, 0 otherwise. Then $a = \min_i x_i = 0$, $b = \max_i x_i = 1$ and $\mu = \frac{\sum_{i=1}^{n} x_i}{n} = q$. For each $i = 1, \ldots, m$, let X_i be the indicator of whether each element of C is in U. Then X_i is a random sample drawn from S without replacement, and $|C \cap U| = \sum_{i=1}^{m} X_i = m\overline{X}$. By Hoeffding's Inequality,

$$\Pr[|C \cap U| \geq (q + \epsilon)m] = \Pr[\overline{X} - \mu \geq \delta] \leq e^{-2m\delta^2}.$$

Therefore, except with probability $e^{-2m\delta^2}$, $|C \setminus U| = m - |C \cap U| > (1 - q - \delta)m$.

Now let $m = \log^{1+\epsilon'} n$ and $\delta = \frac{1-q}{2}$. We have that $|C \setminus U| > \left(\frac{1-q}{2}\right) m$ except with probability

$$e^{-2\left(\frac{1-q}{2}\right)^2 \log^{1+\epsilon'} n} = \frac{1}{n^{c \log^{\epsilon'} n}},$$

where $c = \frac{1}{2}(1-q)^2 \log e$.

Finally, let $q = \frac{1}{2} - \epsilon$ and $\delta = \epsilon$. We have that $|C \setminus U| > \left(1 - \left(\frac{1}{2} - \epsilon\right) - \epsilon\right) m = \frac{1}{2}m$ except with probability $\frac{1}{n^{c' \log^{\epsilon'} n}}$, where $c' = 2\epsilon^2 \log e$.

B Semi-Honest Adaptive Adversaries with Erasures (Cont'd)

Theorem 8 (Generalized Chernoff Bound [PS97]). *Let $X_1, \ldots, X_n$ be Boolean random variables such that, for some $0 \leq \delta \leq 1$, we have that, for every subset $S \subseteq [n]$, $\Pr[\vee_{i \in S} X_i = 1] \leq \delta^{|S|}$. Then for any $0 \leq \delta \leq \gamma \leq 1$, $\Pr[\sum_{i=1}^{n} X_i \geq \gamma n] \leq e^{-nD(\gamma||\delta)}$, where $D(\cdot||\cdot)$ is the relative entropy function satisfying $D(\gamma||\delta) \geq 2(\gamma - \delta)^2$.*

C Semi-Honest Adaptive Adversaries with Erasures — Impossibility

Theorem 4. *Assume that the adversary is adaptive and the parties might erase messages as above. 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 a constant $\epsilon > 0$.*

Proof. Assume towards contradiction that for some constant $\epsilon = O(1)$ a protocol sublinear communication complexity protocol Π exists which tolerates any $t = (1 - \sqrt{0.5} + \epsilon)n$. Without loss of generality we will assume that in Π only a single party speaks in each round. Indeed, any Π can be transformed to a protocol Π' with the above structure by assigning to each party in Π, in a round robin fashion, a round in which only this party might speak; clearly, if Π is secure then $\Pi's$ is also secure.

Consider the following adversary $\mathcal{A}$: At the beginning of the protocol, the adversary $\mathcal{A}$ corrupts each of the n servers independently with probability $1 - \sqrt{0.5}$ and corrupts one of the two clients, say c randomly; 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 in 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.)

By a Chernoff bound it is easy to see that for any constant ϵ : $\Pr[|C_0| > (1 - \sqrt{0.5} + \epsilon)n]$ is negligible, and, because $|C| = |C_0| + o(n)$, $\Pr[|C| > (1 - \sqrt{0.5} + \epsilon)n] = \mu(n)$ for some negligible function $\mu(n)$. (We refer to the end of the proof for an argument of how impossibility against such an adversary can be used to prove impossibility against a $(1 - \sqrt{0.5} + \epsilon)n$-bounded adversary.)

The above adversary jointly with the protocol execution define a random experiment that assigns to each subset $T \subseteq \mathcal{P}$ of parties a probability that (exactly) this set T is the set of *fully corrupted* active parties, i.e., every party in T (and only those) sends a message in the protocol and is already corrupted at the point when he sends his first protocol message. The corresponding experiment can be described as follows:

1. The adversary chooses the parties in C_0 according to the above distribution.
2. The protocol starts executing; let $\hat{e}_1, \hat{e}_2, \ldots,$ denote the set of edges used for communication among the servers in the protocol execution, i.e., where if in round ρ in which s_i send a message to s_j, then $\hat{e}_\rho = (s_i, s_j)$. Let $\boldsymbol{E} = (e_1, e_2, \ldots, e_\ell)$ be the subsequence of this edge sequence where every edge is only included the first time it is used (in either direction, i.e., if $\hat{e}_\rho = (s_i, s_j)$, then every $\hat{e}_{\rho'} \in \{(s_i, s_j), (s_j, s_i)\}$ with $\rho' > \rho$ is removed). For each $e_i = (s_i, s_j) \in \boldsymbol{E}$, if any of it endpoints p_i or p_j is corrupted then the adversary corrupts also the other endpoint.
3. The random variable C is the set of corrupted parties at that the end of the above experiment. For the set of eventually active parties $\bar{\mathcal{S}}$ (i.e., parties the send or receive a message at some point in the protocol), we can also define the random variable $C|_{\bar{\mathcal{S}}}$ corresponding to the set of eventually active parties $\bar{\mathcal{S}}$ that are fully corrupted at the end of the protocol.[13]

Clearly the entire view of all parties in C (resp. $C|_{\bar{\mathcal{S}}}$) is included in the adversary's view.

Because the adversary is semi-honest, hence the set of corrupted parties has no effect in the protocol's execution, and the initial set C_0 is chosen by $\mathcal{A}$ independently of the protocol execution, the above probability distribution on $C|_{\bar{\mathcal{S}}}$ can be obtained by an equivalent experiment in which first the edges are defined (computing according to the protocol's inputs and randomness, and then the adversary makes his corruption of C_0 independently, and following the protocol. Therefore, in the remaining of the proof we will make the assumption that the sequence $\boldsymbol{E}$ of first used edges is fixed and show that the above adversary cannot be tolerated.

Towards this direction, let $\boldsymbol{E} = (e_1, e_2, \ldots, e_\ell)$ be the sequence/vector of (disjoint) communication edges that are used in the protocol. In slight abuse of notation we might use a vector as a set (but of course not vice versa). For example, for a set E we will write $E \subseteq \boldsymbol{E}$ to denote the fact that E consists of edges from $\boldsymbol{E}$. By definition this sequence spans the entire eventually active server set $\bar{\mathcal{S}}$.

Edge traversing algorithm. For each edge $e_i \in \boldsymbol{E}$ we will compute a set E_i which is the biggest reachable set when only using edges in the sequence the appear in $\boldsymbol{E}$. Concretely, consider for each edge e_i, the set E_i of edge that is traversed starting with e_i via the following algorithm where $\boldsymbol{E} = (e_1, \ldots, e_\ell)$:

- For each $i = 1, \ldots, \ell$ do
  - Set $E_i = e_i$
  - For each $j = i + 1 \ldots, \ell$ do
    - If there exists a path from e_i (i.e., from any of the vertices of e_i) to e_j (i.e., from any of the vertices of e_j) that uses only edges in E_i, then update $E_i = E_i \cup \{e_j\}$

Denote by $\boldsymbol{E}^o$ the vector $\boldsymbol{E}^o = (E_1, \ldots, E_\ell)$.

Definition 1 (edge corruption). *Given a sequence of edges $\boldsymbol{E} = (e_1, \ldots, e_\ell)$ as above that are used in the protocol in this order, we say that an adversary corrupts an edge $e_i = (p_1, p_2) \in \boldsymbol{E}$ if at the at least one of the parties p_1 or p_2 has been corrupted before the edge e_i is used.*

Probabilistic Adversary Structures. For a give edge sequence $\boldsymbol{E}$, our above defined adversary $\mathcal{A}$ induces a *probabilistic adversary structure* $\Pr_{\mathcal{A}}$, i.e., a probability distribution on subsets of the

[13] Note that as we allow erasures, post execution corruption of servers is irrelevant.

active party set $\bar{\mathcal{P}} = \bar{\mathcal{S}} \cup \{c_1, c_2\}$, which assigns to each $T \subseteq \bar{\mathcal{P}}$ probability $\Pr_{\mathcal{A}}(T)$ of T being corrupted by $\mathcal{A}$ at the end of the protocol. It also induces a probabilistic edge-corruption adversary (structure) $\Pr_{\mathcal{A}^E}$, on subsets of $\{e_1, \ldots, e_\ell\}$, where for each $E \subset \{e_1, \ldots, e_\ell\}$, $\Pr_{\mathcal{A}^E}(E)$ is the probability that (exactly) the edges in E get corrupted by $\mathcal{A}$ (according to Definition 1).

Definition 2 (intolerable adversary structure). *We say that a probabilistic adversary structure* $\Pr_{\mathcal{A}}$ *is* intolerable *if there exists no secure OT protocol in the* $(2, n)$*-client/server model tolerating a* $\Pr_{\mathcal{A}}$*-adversary, i.e., an adversary that corrupts a set* T *of parties with probability* $\Pr_{\mathcal{A}}(T)$*.*

Definition 3 (intolerable edge adversary structure). *Let* $\Pr_{\mathcal{A}^E}$ *be an edge corruption probabilistic structure on the set of edges between servers. Let also* $\Pr_{\mathcal{A}}$ *be the induced probabilistic adversary structure on the set of servers that assigns to each* $S' \subseteq S$ *probability according to the following experiment:*

1. $E \leftarrow \Pr_{\mathcal{A}^E}$
2. $P = \{s_i \mid \text{ for } s_j \in \mathcal{P} \setminus \{c_1, c_2\}\{(s_i, s_j) \cup (s_j, s_i)\} \cap E \neq \emptyset\}$

Denote by $\Pr_{\bar{\mathcal{A}}}$ *the following extension of* $\Pr_{\mathcal{A}}$ *to the full party set* $\mathcal{P} = \{c_1, c_2\} \cup S$*: Corrupt a server subset* T *with probability* $\Pr_{\mathcal{A}}(T)$*, and additionally corrupt one of the two clients with probability* $1/2$*. We say that* $\Pr_{\mathcal{A}^E}$ *is intolerable if and only if* $\Pr_{\bar{\mathcal{A}}}$ *is intolerable.*

Definition 4 (exact (unique) cover). *Let* Q *be a set,* $\mathcal{T} = \{T_1, \ldots, T_\ell\}$ *be such that each* $T_i \subseteq Q$ *and let* $T \subseteq Q$*.*

- *We say that* T is covered by $\mathcal{T}$ *if* $\exists \mathcal{I} \subseteq [\ell] : T \subseteq \cup_{i \in \mathcal{I}} T_i$*.*
- *We say that* T is exactly covered by $\mathcal{T}$*, and denote it as* $T \lhd \mathcal{T}$*, if* $\exists \mathcal{I} \subseteq [\ell] : T = \cup_{i \in \mathcal{I}} T_i$*. In this case, a set* $\mathcal{EC}_{\mathcal{T}}(T)$ *such that* $\mathcal{EC}_{\mathcal{T}}(T) = \{T_i \in \mathcal{T} \mid i \in \mathcal{I}\}$ *is called an* exact cover of T by $\mathcal{T}$*. We also denote by* $T \not\lhd \mathcal{T}$ *the fact that* T *is not covered by* $\mathcal{T}$*.*
- *We say that* T is uniquely exactly covered by $\mathcal{T}$ *if* $\exists! \mathcal{I} \subseteq [\ell] : T = \cup_{i \in \mathcal{I}} T_i$*.*

Lemma 5. *Let* $E \subseteq \boldsymbol{E}$*. If* $E \not\lhd \boldsymbol{E}^o$ *then* $\Pr_{\mathcal{A}^E}(E) = 0$*.*

Proof. For the above vector $\boldsymbol{E}^o$ and the adversary $\mathcal{A}$ let $E \subseteq \boldsymbol{E}$ be such that $\Pr_{\mathcal{A}}(E) > 0$. For any e_i, if $e_i \in E$ and E gets corrupted, it means that its endpoints had been corrupted (at the latest) by the round in which e_i was first used and therefore all nodes in E_i will also be corrupted by our adversary. Hence, for any E with $\Pr_{\mathcal{A}}(E) > 0$ we have $E \subseteq \cup_{e_i \in E} E_i$. Furthermore, because e_i might *only* be included in E during the round when it is first used all edges that are in E_i will end up being corrupted too, hence $E_i \subseteq E$ for all $e_i \in E$ which means that $\cup_{e_i \in T} E_i \subseteq T$. Hence, if $\Pr_{\mathcal{A}}(E) > 0$ then $E = \cup_{e_i \in E} E_i$. □

Definition 5 (maximal set). *A set* E_i *is called a* maximal set *in the sequence* $\boldsymbol{E}^o$ *iff* $\forall j \neq i : E_i \not\subseteq E_j$*. We denote by* $\boldsymbol{E}_{max}$ *the set of maximal sets.*

Definition 6 ((exact) disjoint edge cover). *For a vector* $\boldsymbol{E} = (E_1, \ldots, E_\ell)$ *of sets of edges on a vertex set* $Q \subseteq \mathcal{P}$*, we say that* $\boldsymbol{E}$ *is a* disjoint edge cover *of* Q *if the following properties hold:*

1. $E' = \cup_{E \in \boldsymbol{E}} E$ *induces a vertex cover on* Q*, i.e.,* $Q \subseteq \{v_i \mid \exists v_j \in \mathcal{P} : \{(v_i, v_j) \cup (v_j, v_i)\} \cap E' \neq \emptyset\}$
2. $\forall E_i \in \boldsymbol{E} : \; E_i \not\subseteq \cup_{E_j \in \boldsymbol{E} \setminus \{E_i\}} E_j$*.*

If property 1 *holds with equality, i.e.,* $Q = \{v_i \mid \exists v_j \in \mathcal{P} : \{(v_i, v_j) \cup (v_j, v_i)\} \cap E' \neq \emptyset\}$*, (and Property 2 holds too) then we say that* $\boldsymbol{E}$ *is an* exact disjoint edge cover of Q*.*

Lemma 6. *Let $\boldsymbol{E} = (E_1, \ldots, E_\ell)$ be a non-empty exact disjoint edge cover of the server set S and $\Pr_{\mathcal{A}^E_{1/2}}$ be the probability distribution over subsets of servers corresponding to the following experiment:*

1. *Set $E = \emptyset$*
2. *For each $i \in [\ell]$ (in any order): Choose a bit b_i with probability $1/2$ and if $b = 1$ set $E := E \cup E_i$*

Output T. Then $\Pr_{\mathcal{A}^E_{1/2}}$ is intolerable.

Proof. Assume that there exists a Π protocol which tolerates the $\Pr_{\mathcal{A}^E_{1/2}}$-adversary (recall, we are in the semi honest model). We show how to use protocol Π to construct a protocol $\hat{\Pi}$ for two parties $\hat{p}_1$ and $\hat{p}_2$ with inputs bits b_1 and b_2, respectively, to compute $b = b_1 \vee b_2$ in the presence of an adversary who corrupts either of the two parties with probability $1/2$. Existence of such a protocol contradicts the impossibility of information-theoretic 2PC from [BGW88].
The protocol $\hat{\Pi}$ works as follows: The parties $\hat{p}_1$ and $\hat{p}_2$ emulate an execution of protocol Π for computing the AND of the bits of the clients where for $i \in \{1, 2\}$ $\hat{p}_i$ plays for client c_i and the virtual servers $\hat{S}$ are emulated as follows: Each party $\hat{p}_i$ emulates the set of servers $\hat{S}_i$, where $\hat{S}_1$ and $\hat{S}_2$ are sampled as follows:

- Initialize $\hat{E}_1 := \hat{E}_2 := \emptyset$.
- For each $i \in [\ell]$ (in any order) the parties choose $\hat{p}_1$ and $\hat{p}_2$ a random bit b_i (recall that they are semi-honest so this is trivial) and if $b = 1$ set $\hat{E}_1 := \hat{E}_1 \cup E_i$, otherwise $\hat{E}_2 := \hat{E}_2 \cup E_i$.
- For $j \in \{1, 2\}$: Set $\hat{S}_j = \{s_k \in S \mid \text{ for } s_q \in S : \{(s_i, s_j) \cup (s_j, s_i)\} \cap \hat{E}_j \neq \emptyset\}$; i.e., the $\hat{S}_j$ is the set of active servers that is covered by $\hat{E}_j$.

We remark that because $\boldsymbol{E} = (E_1, \ldots, E_\ell)$ is a non-empty exact edge cover of the (virtual) server set $\hat{S}$, $\hat{S}_1 \cup \hat{S}_2 = S$. Furthermore, we note that some servers will be in $\hat{S}_1 \cap \hat{S}_2$; such virtual servers are jointly emulated by having having both $\hat{p}_1$ and $\hat{p}_2$ choose his coins and exchanging any messages this server is supposed to exchange in the protocol.

Let $\mathcal{A}_1$ be a semi-honest adversary that corrupts one of the two clients in the above protocol randomly and outputs its view. This $\mathcal{A}_1$ corrupts the (virtual) parties with the same probability as a $\Pr_{\mathcal{A}^E_{1/2}}$-adversary corrupts the real parties. Indeed, $\mathcal{A}_1$ corrupts each of the two parties with probability $1/2$ and additionally, each set covered by the edges in each E_j with probability $1/2$ independent of whether or not the other sets are corrupted, which identical to how the $\Pr_{\mathcal{A}^E_{1/2}}$-adversary corrupts parties in the protocol. Thus, the simulator which is assumed to exists by the assumption that Π is secure can be used to simulate $\mathcal{A}_1$. But existence of such as simulator directly contradicts the classical impossibility results for two-party computation of the AND gate where either of the parties can be corrupted [BGW88]. □

Definition 7 (dominating edge corruption structure). *Let $\Pr_{\mathcal{A}^E_1}$ and $\Pr_{\mathcal{A}^E_2}$ be two probabilistic adversary structures on an edge set $\boldsymbol{E}$. We say that $\Pr_{\mathcal{A}^E_1}$* dominates *$\Pr_{\mathcal{A}^E_2}$ and denote it as $\Pr_{\mathcal{A}^E_1} \geq \Pr_{\mathcal{A}^E_2}$, if there exists a probabilistic mapping $F_{edge} : 2^{\boldsymbol{E}} \to 2^{\boldsymbol{E}}$ such that the following properties hold: (1) for every $E \subseteq \boldsymbol{E} : \Pr[F_{edge}(E) \subseteq E] = 1$; (2) Consider the random variable $\mathcal{E}_2$ defined via the following experiment:*

1. *$E_1 \leftarrow \Pr_{\mathcal{A}^E_1}$*
2. *$E_2 \leftarrow F_{edge}(E_1)$*

Then for each $E \subseteq \boldsymbol{E}$*:* $\Pr_{\mathcal{E}_2}(E) = \Pr_{\mathcal{A}_2^E}(E)$.

Definition 8 (dominating party corruption structure). *Let* $\Pr_{\mathcal{A}_1}$ *and* $\Pr_{\mathcal{A}_2}$ *be two probabilistic adversary structures on the party set* $\mathcal{P}$*. We say that* $\Pr_{\mathcal{A}_1}$ dominates $\Pr_{\mathcal{A}_2}$ *and denote it as* $\Pr_{\mathcal{A}_1} \geq \Pr_{\mathcal{A}_2}$*, if there exists a probabilistic mapping* $F : 2^{\mathcal{P}} \to 2^{\mathcal{P}}$ *such that the following properties hold: (1) for every* $T \subseteq \mathcal{P} : \Pr[F(T) \subseteq T] = 1$*; (2) Consider the random variable* $\mathcal{T}_2$ *defined via the following experiment:*

1. $T_1 \leftarrow \Pr_{\mathcal{A}_1}$
2. $T_2 \leftarrow F(T_1)$

Then for each $T \subseteq \mathcal{P}$*:* $\Pr_{\mathcal{T}_2}(T) = \Pr_{\mathcal{A}_2}(T)$.

Lemma 7. *Let* $\Pr_{\mathcal{A}_1^E}$ *and* $\Pr_{\mathcal{A}_2^E}$ *be probabilistic edge-corruption structures ever sets of edges of the server-set* S*, and* $\Pr_{\mathcal{A}_1}$ *and* $\Pr_{\mathcal{A}_2}$ *be the induced probabilistic server-corruption structures, respectively. If* $\Pr_{\mathcal{A}_1^E} \geq \Pr_{\mathcal{A}_2^E}$ *then* $\Pr_{\mathcal{A}_1} \geq \Pr_{\mathcal{A}_2}$.

Proof. Let $F_{\text{edge}}(\cdot)$ be the forgetting mapping that is guaranteed to exists by the assumption $\Pr_{\mathcal{A}_1^E} \geq \Pr_{\mathcal{A}_2^E}$. Set $F(T) = \{s_i \mid \text{ for } s_j \in \mathcal{P} \setminus \{c_1, c_2\}\{(s_i, s_j) \cup (s_j, s_i)\} \cap F_{\text{edge}}(E) \neq \emptyset\}$. By inspection of the experiments it is easy to verify that the output distribution of the experiment in Definition 8 is identical with the output distribution in Definition 3 where the edge-sampling step (Step 1) is replaced by sampling according to the experiment in Definition 7. □

Lemma 8. *Let* $\Pr_{\mathcal{A}_1^E}$ *and* $\Pr_{\mathcal{A}_2^E}$ *be probabilistic edge-corruption structures over the set of edges among servers* S*. If* $\Pr_{\mathcal{A}_2^E} \geq \Pr_{\mathcal{A}_1^E}$ *and* $\Pr_{\mathcal{A}_1{}^E}$ *is intolerable, then* $\Pr_{\mathcal{A}_2{}^E}$ *is also intolerable.*

Proof. Let $\Pr_{\mathcal{A}_1}$ and $\Pr_{\mathcal{A}_2}$ be the induced server corruption structures. By Lemma 7 it suffices to prove that if $\Pr_{\mathcal{A}_2} \geq \Pr_{\mathcal{A}_1}$ and the adversary $\mathcal{A}_1$ that corrupts one of the two clients with probability $1/2$ and additionally corrupts a set T of servers with probability $\Pr_{\mathcal{A}_1}(T)$ is intolerable, then so is the adversary $\mathcal{A}_2$ that corrupts one of the two clients with probability $1/2$ and additionally corrupts a set T of servers with probability $\Pr_{\mathcal{A}_2}(T)$.

Assume towards contradiction that there exist a protocol Π which is secure against such an $\mathcal{A}_2$. (Recall that we are in the semihonest setting, hence wlog we can assumr that the adversaries only defer in the set of parties they are corrupt and they output their entire protocol view.) This means that there exists a simulator σ such that for every such $\Pr_{\mathcal{A}_2}$-adversary $\mathcal{A}_2$

$$\textsc{View}_{\mathcal{A}_2,\Pi} \approx \textsc{View}_{\sigma,f}. \tag{8}$$

We prove that Π is also secure against any $\Pr_{\mathcal{A}_1}$-adversary leading to a contradiction. Let $\mathcal{A}_1$ be an adversary. Define the adversary $\mathcal{A}_3$ that samples the set of corrupted parties as follows: corrupts one of the two client randomly and additionally corrupts a set T of servers as follows: $\mathcal{A}_3$ samples $T_2 \leftarrow \Pr_{\mathcal{A}_2}$, and computes $T \leftarrow F(T_2)$, where F is the mapping that is guaranteed to exist by $\Pr_{\mathcal{A}_2} \geq \Pr_{\mathcal{A}_1}$. If follows immediately from the domination definition that

$$\textsc{View}_{\mathcal{A}_3,\Pi} \equiv \textsc{View}_{\mathcal{A}_1,\Pi} \tag{9}$$

Now consider the simulator σ' that receives the set T of (finally) corrupted parties by σ and applies to this set F, i.e., σ' corrupts $F(T)$. By definition of $\mathcal{A}_3$ and Equation 8 , because σ applied the same transformation on its view as $\mathcal{A}_3$, we get

$$\text{VIEW}_{\sigma',\Pi} \approx \text{VIEW}_{\mathcal{A}_3,\Pi} \tag{10}$$

But Equations 9 and 10 imply that σ' is a good simulator from $\mathcal{A}_1$ which contradicts the assumption that $\mathcal{A}_1$ is intolerable.

Lemma 9. $\forall E_i \in \boldsymbol{E}_{max} :\ E_i \not\subseteq \cup_{E_j \in \boldsymbol{E}^o \setminus \{E_i\}} E_j$.

Proof. The claim follows from the fact that for every maximal set $E_i \in \boldsymbol{E}_{max}$, the edge e_i (i.e., the first edge added to this set in the above algorithm) is not included in any $E_j \in \boldsymbol{E}^o$ with $E_j \neq E_i$. Indeed, if for some E_j we have $e_i \in E_j$ then the above traversing will yield $E_j \supseteq E_i$ which contradicts the maximality of E_i. □

Lemma 10. *For any edge set $E \in \cup_{E_i \in \boldsymbol{E}^o} E_i$, let $\texttt{Corr}(E)$ denote the event that (at least) E gets corrupted by $\mathcal{A}$. For every $E_i \in \boldsymbol{E}_{max}$ and every $E_j \in \boldsymbol{E}^o \setminus E_i$: $\Pr_{\mathcal{A}^E}(\texttt{Corr}(E_i)|\texttt{Corr}(E_j)) = \Pr_{\mathcal{A}^E}(\texttt{Corr}(E_i)) = 1/2$.*

Proof. Recall that we denote by C_0 the set of parties that are initially corrupted by $\mathcal{A}$. Because $\boldsymbol{E}_{max}$ is a maximal set, Lemma 9 implies that $\forall E_j \in \boldsymbol{E}^o :\ E_i \not\subseteq \cup_{E_j \in \boldsymbol{E}^o \setminus \{E_i\}} E_j$. In fact, it is easy to verify that for the first edge e_i in E_i, $e_i \notin \cup_{E_j \in \boldsymbol{E}^o \setminus \{E_i\}} E_j$. Hence the only way that e_i gets corrupted is if s_1 or s_2 is in the initial set of corrupted parties. I.e., If $e_i = (s_1, s_2)$ then $\Pr_{\mathcal{A}^E}(\texttt{Corr}(E_i)) \geq \Pr_{\mathcal{A}^E}(\texttt{Corr}(\{e_i\})) = \Pr_{\mathcal{A}^E}(\{s_1, s_2\} \cap C_0 \neq \emptyset)$. Moreover, by the definition of $\mathcal{A}$, if e_i gets corrupted then the entire E_i will get corrupted; i.e., $\Pr_{\mathcal{A}^E}(\texttt{Corr}(E_i)) \leq \Pr_{\mathcal{A}^E}(\texttt{Corr}(\{e_i\})) = \Pr_{\mathcal{A}^E}(\{s_1, s_2\} \cap C_0 \neq \emptyset)$. Therefore, the probability that E_i will get corrupted equals the probability that edge e_i has been corrupted during the initial (random) corruption step of $\mathcal{A}$. But this probability is $1/2$ independent of what other vertexes or edges get corrupted. □

Lemma 11. *Let $E \lhd \boldsymbol{E}_{max}$. Then E is uniquely exactly covered by $\boldsymbol{E}_{max}$.*

Proof. Let $\hat{\mathbf{E}} = (\hat{E}_1, \ldots, \hat{E}_m)$ be an exact cover of E by $\boldsymbol{E}_{max}$. Such a cover in guaranteed to exists from the assumption that $E \lhd \boldsymbol{E}_{max}$. It suffices to prove uniqueness. Assume, towards contradiction, that there exists another exact cover of E by $\boldsymbol{E}_{max}$, and denote it by $\mathbf{E}' = (E'_1, \ldots, E'_q)$. This means that

$$\cup_{i=1,\ldots,m} \hat{E}_i = \cup_{i=1,\ldots,q} E'_i$$

But as argued in Lemma 10, the first edge of each $E \in \boldsymbol{E}_{max}$ is not included in any $E'' \in \boldsymbol{E}_{max} \setminus E$. Hence, if $E_i = (e_i, \ldots)$ is a set that is not included in both the above covers, then the edge e_i (i.e., the first edges added in E_i is our graph traversal) cannot be in both sides of the above equation which leads to a contradiction. □

Lemma 12. *Let $\Pr_{\mathcal{A}^E}$ denote the probabilistic edge-corruption structure that is induced by adversary $\mathcal{A}$ when $\boldsymbol{E}$ is the sequence of edges. Then $\Pr_{\mathcal{A}^E} \geq \Pr_{\mathcal{A}^E_{1/2}}$.*

Proof. Let $E \subseteq \cup_{E_i \in \boldsymbol{E}^o} E_i$ be an edge set. We know the following

- If $E \not\lhd \boldsymbol{E}^o$ then $\Pr_{\mathcal{A}^E}(E) = 0$ (Lemma 5)
- If $E \lhd \boldsymbol{E}^o$ but $E \not\lhd \boldsymbol{E}_{max}$, then $\Pr_{\mathcal{A}^E}(E) \geq 0$

– If $E \lhd \boldsymbol{E}_{max}$ and denote by $\hat{\mathbf{E}} = (E_{i_1}, \ldots, E_{i_m})$ the unique exact cover of E by $\boldsymbol{E}_{max}$ (which is guaranteer to exist by Lemma 11). Then

$$\Pr_{\mathcal{A}^E}(E) = \Pr_{\mathcal{A}^E}\left(\left(\bigwedge_{E\in\hat{\mathbf{E}}} \mathtt{Corr}(E)\right) \wedge \left(\bigwedge_{E\in\boldsymbol{E}_{max}\setminus\hat{\mathbf{E}}} \overline{\mathtt{Corr}(E)}\right)\right) \tag{11}$$

Indeed, because $\hat{\mathbf{E}}$ exact cover of E we have that

$$\Pr_{\mathcal{A}^E}(E) \geq \Pr_{\mathcal{A}^E}\left(\left(\bigwedge_{E\in\hat{\mathbf{E}}} \mathtt{Corr}(E)\right) \wedge \left(\bigwedge_{E\in\boldsymbol{E}_{max}\setminus\hat{\mathbf{E}}} \overline{\mathtt{Corr}(E)}\right)\right)$$

Furthermore, as in Lemma 10, it is easy to verify that because each edge e_{i_j} (i.e., the first edge in E_{i_j}) can be covered only when E_{i_j} is entirely corrupted (i.e., when $\mathtt{Corr}(E_{i_j}$ occurs), we have that

$$\Pr_{\mathcal{A}^E}(E) \leq \Pr_{\mathcal{A}^E}\left(\left(\bigwedge_{E\in\hat{\mathbf{E}}} \mathtt{Corr}(E)\right) \wedge \left(\bigwedge_{E\in\boldsymbol{E}_{max}\setminus\hat{\mathbf{E}}} \overline{\mathtt{Corr}(E)}\right)\right).$$

We next define a probabilistic forget mapping $F : 2^{\cup_{E\in\boldsymbol{E}^o}} \to 2^{\cup_{E\in\boldsymbol{E}^o}}$ as follows: Given input any $E \subseteq \cup_{E\in\boldsymbol{E}^o}$, $F(E)$ computes its output as follows:

– If $E \not\lhd \boldsymbol{E}^o$ then $F(E) = E$
– If $E \lhd \boldsymbol{E}^o$ but $E \not\lhd \boldsymbol{E}_{max}$ then $F(E) = \emptyset$
– If $E \lhd \boldsymbol{E}_{max}$ then $F(E) = E$.

As in Definition 7, denote by $\Pr_{\mathcal{E}'}$ the distribution of the random variable $\mathcal{E}'$ of the output E' of the following experiment.

1. $E \leftarrow \Pr_{\mathcal{A}^E}$
2. $E' \leftarrow F(E)$

It is straightforward to verify that F satisfies the requirements of Definition 7, hence

$$\Pr_{\mathcal{A}^E} \geq \Pr_{\mathcal{E}'} \tag{12}$$

Furthermore $\Pr_{\mathcal{E}}$ is the following distribution:

– For any $E \not\lhd \boldsymbol{E}_{max}$:

$$\Pr_{\mathcal{E}'}(E) = \Pr_{\hat{\mathcal{A}}^E|\mathcal{A}^E}(E|E \not\lhd \boldsymbol{E}^o)\Pr_{\mathcal{A}^E}(E \not\lhd \boldsymbol{E}^o) + \Pr_{\hat{\mathcal{A}}^E|\mathcal{A}^E}(E|E \lhd \boldsymbol{E}^o)\Pr_{\mathcal{A}^E}(E \lhd \boldsymbol{E}^o) = 0$$

– For any $E \lhd \boldsymbol{E}_{max}$:

$$\Pr_{\mathcal{E}'}(E) = \Pr_{\mathcal{A}^E}\left(\left(\bigwedge_{E\in\hat{\mathbf{E}}} \mathtt{Corr}(E)\right) \wedge \left(\bigwedge_{E\in\boldsymbol{E}_{max}\setminus\hat{\mathbf{E}}} \overline{\mathtt{Corr}(E)}\right)\right).$$

Since for different $E \in \boldsymbol{E}_{max}$ the events $\mathtt{Corr}(E)$ are independent and each has probability $1/2$ (Lemma 10), $\Pr_{\mathcal{E}'}$ is identical to $\Pr_{\mathcal{A}^E_{1/2}}$. Therefore, Equation 12 implies that

$$\Pr_{\mathcal{A}^E} \geq \Pr_{\mathcal{A}^E_{1/2}}$$

□

The following corollary can be easily derived by combining the above lemma with Lemma 8 because $\Pr_{\mathcal{A}^E_{1/2}}$ is intolerable (as proved in Lemma 6).

Corollary 3. $\Pr_{\mathcal{A}^E}$ *is intolerable.*

The above Corollary shows that the adversary $\mathcal{A}$ cannot be simulated. However, by definition $\mathcal{A}$ might corrupt more than $(1 - \sqrt{0.5} + \epsilon)n$ of the servers already at the initial step (since each server is added or not to C_0 independently). Thus the intolerability of $\mathcal{A}$ does not suffice. For this reason we consider the following adversary $\mathcal{A}'$: $\mathcal{A}$ works exactly as $\mathcal{A}$ with the only difference that if at any point it corrupts more than $(1 - \sqrt{0.5} + \epsilon)n$ parties it aborts. A direct application of the Chernoff bound implies that the probability that $\mathcal{A}'$ aborts is negligible in n. Therefore, because conditioned on non-aborting the view of $\mathcal{A}'$ is identical to the view of $\mathcal{A}'$, any good simulator for $\mathcal{A}'$ is also a good simulator for $\mathcal{A}$. But the intolerability of $\mathcal{A}$ trivially implies intolerability of $\mathcal{A}'$. □